



Assignment-01

Roll No: 123M1H010

Name of Student: Harshal Sanjay Bhamare

Submission Date: 24 / 08 / 24

1. Write a program to print Hello World.

Solution:

```
package com.example.forpractice;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World"
        android:textSize="30sp" />
</RelativeLayout>
```

Output:



2. Write an android application that will accept two numbers from the users and will print the addition as output in TextView and Toast.

Solution:

```
package com.example.forpractice;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        EditText number1EditText = findViewById(R.id.number1);
        EditText number2EditText = findViewById(R.id.number2);
        Button addButton = findViewById(R.id.addButton);
        TextView resultTextView = findViewById(R.id.resultTextView);

        addButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String num1Str = number1EditText.getText().toString();
                String num2Str = number2EditText.getText().toString();

                if (!num1Str.isEmpty() && !num2Str.isEmpty()) {
                    double num1 = Double.parseDouble(num1Str);
                    double num2 = Double.parseDouble(num2Str);
                    double sum = num1 + num2;

                    resultTextView.setText("Sum: " + sum);
                    Toast.makeText(getApplicationContext(), "Sum is " + sum, Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

```
        Toast.makeText(MainActivity.this, "Sum: " + sum,  
Toast.LENGTH_LONG).show();  
    } else {  
        Toast.makeText(MainActivity.this, "Please enter both  
numbers", Toast.LENGTH_SHORT).show();  
    }  
}  
});  
}  
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <EditText
        android:id="@+id/number1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter first number"
        android:inputType="number" />

    <EditText
        android:id="@+id/number2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter second number"
        android:inputType="number"
        android:layout_below="@id/number1"
        android:layout_marginTop="16dp"/>

    <Button
        android:id="@+id/addButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Add"
        android:layout_below="@id/number2"
        android:layout_marginTop="16dp"
        android:layout_centerHorizontal="true"/>

    <TextView
        android:id="@+id/resultTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:layout_below="@id/addButton"
        android:layout_marginTop="16dp"
        android:layout_centerHorizontal="true"/>

</RelativeLayout>
```

Output:



3. Write an android application that will demonstrate an Android Activity Life Cycle and print output in the logcat.

Solution:

```
package com.example.forpractice;

import android.os.Bundle;
import android.util.Log;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private static final String TAG = "Lifecycle";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(TAG, "onCreate called");
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.d(TAG, "onStart called");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d(TAG, "onResume called");
    }

    @Override
    protected void onPause() {
        super.onPause();
    }
}
```

```

        Log.d(TAG, "onPause called");
    }

    @Override
    protected void onStop() {
        super.onStop();
        Log.d(TAG, "onStop called");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d(TAG, "onRestart called");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d(TAG, "onDestroy called");
    }
}

```

Output:

2024-08-24 18:58:55.439 7177-7177 Lifecycle
 2024-08-24 18:58:55.525 7177-7177 Lifecycle
 2024-08-24 18:58:55.530 7177-7177 Lifecycle
 2024-08-24 19:00:30.643 7177-7177 Lifecycle
 2024-08-24 19:00:31.797 7177-7177 Lifecycle
 2024-08-24 19:00:31.842 7177-7177 Lifecycle

com.example.forpractice	D onCreate called
com.example.forpractice	D onStart called
com.example.forpractice	D onResume called
com.example.forpractice	D onPause called
com.example.forpractice	D onStop called
com.example.forpractice	D onDestroy called

4. Write an android application that will display components in GridLayout.

Solution:

```

<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="2"
    android:rowCount="3"
    android:padding="16dp"
    android:alignmentMode="alignMargins"
    android:useDefaultMargins="true">

    <Button
        android:text="Button 1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_columnWeight="1"
        android:layout_row="0"
        android:layout_column="0"/>

    <Button
        android:text="Button 2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_columnWeight="1"
        android:layout_row="1"
        android:layout_column="0"/>

    <Button
        android:text="Button 3"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_columnWeight="1"
        android:layout_row="2"
        android:layout_column="0"/>

```

```
        android:layout_row="0"
        android:layout_column="1"/>

    <Button
        android:text="Button 3"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_columnWeight="1"
        android:layout_row="1"
        android:layout_column="0"/>

    <Button
        android:text="Button 4"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_columnWeight="1"
        android:layout_row="1"
        android:layout_column="1"/>

    <Button
        android:text="Button 5"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_columnWeight="1"
        android:layout_row="2"
        android:layout_column="0"/>

    <Button
        android:text="Button 6"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_columnWeight="1"
        android:layout_row="2"
        android:layout_column="1"/>

</GridLayout>
```

Output:



5. Write an android application that will display components in TableLayout.

Solution:

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:stretchColumns="1">

    <TableRow>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Row 1, Col 1"
            android:padding="8dp"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Row 1, Col 2"
            android:padding="8dp"/>

        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 1"
            android:padding="8dp"/>
    </TableRow>

    <TableRow>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Row 2, Col 1"
            android:padding="8dp"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Row 2, Col 2"
            android:padding="8dp"/>

        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 2"
            android:padding="8dp"/>
    </TableRow>

    <TableRow>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Row 3, Col 1"
            android:padding="8dp"/>

        <TextView
            android:layout_width="wrap_content"
```

```

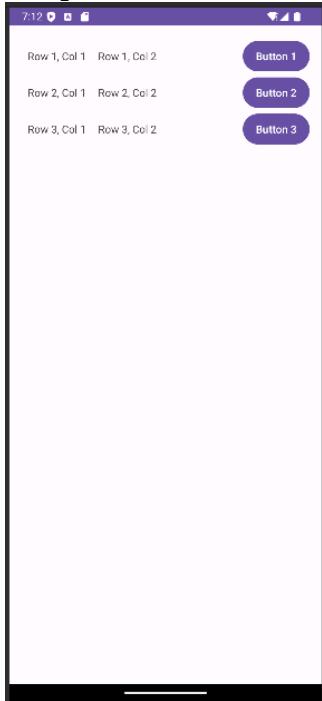
        android:layout_height="wrap_content"
        android:text="Row 3, Col 2"
        android:padding="8dp"/>

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 3"
        android:padding="8dp"/>
</TableRow>

</TableLayout>

```

Output:



6. Write an android application which will change the background color of an activity on switching the state of the ToggleButton.

Solution:

```

package com.example.forpractice;

import android.graphics.Color;
import android.os.Bundle;
import android.widget.RelativeLayout;
import android.widget.ToggleButton;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

```

```
// Find the ToggleButton and RelativeLayout by their IDs
ToggleButton toggleButton = findViewById(R.id.toggleButton);
final RelativeLayout layout = findViewById(R.id.main_layout); // Add
an id to RelativeLayout

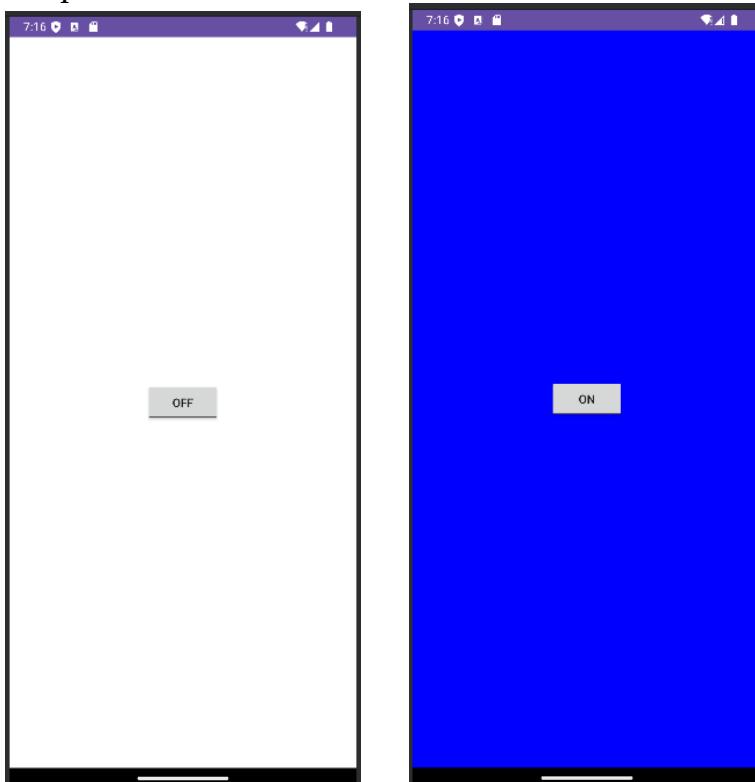
// Set an OnCheckedChangeListener on the ToggleButton
toggleButton.setOnCheckedChangeListener((buttonView, isChecked) -> {
    if (isChecked) {
        // If ToggleButton is ON, change background color to Blue
        layout.setBackgroundColor(Color.BLUE);
    } else {
        // If ToggleButton is OFF, change background color to White
        layout.setBackgroundColor(Color.WHITE);
    }
});
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <ToggleButton
        android:id="@+id/toggleButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textOff="OFF"
        android:textOn="ON"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

Output:



7. Write an android application which will display the menu with the price of the menu item. Once the final order is completed display the total bill amount to the user in Toast. Use CheckBox for creating the menu items.

Solution:

```
package com.example.forpractice;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private final double BURGER_PRICE = 5.00;
    private final double FRIES_PRICE = 2.00;
    private final double SODA_PRICE = 1.50;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Find CheckBox and Button by their IDs
        CheckBox checkboxItem1 = findViewById(R.id.checkbox_item1);
        CheckBox checkboxItem2 = findViewById(R.id.checkbox_item2);
        CheckBox checkboxItem3 = findViewById(R.id.checkbox_item3);
        Button buttonCalculate = findViewById(R.id.button_calculate);

        // Set an OnClickListener on the Calculate Button
        buttonCalculate.setOnClickListener(v -> {
            double total = 0.0;

            // Check which CheckBoxes are selected and calculate total
            if (checkboxItem1.isChecked()) {
                total += BURGER_PRICE;
            }
            if (checkboxItem2.isChecked()) {
                total += FRIES_PRICE;
            }
            if (checkboxItem3.isChecked()) {
                total += SODA_PRICE;
            }

            // Display the total bill in a Toast
            Toast.makeText(MainActivity.this, "Total Bill: $" + total,
Toast.LENGTH_SHORT).show();
        });
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">
```

```
<CheckBox
    android:id="@+id/checkbox_item1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Burger - $5.00"
    android:layout_alignParentStart="true"
    android:layout_marginTop="20dp"/>

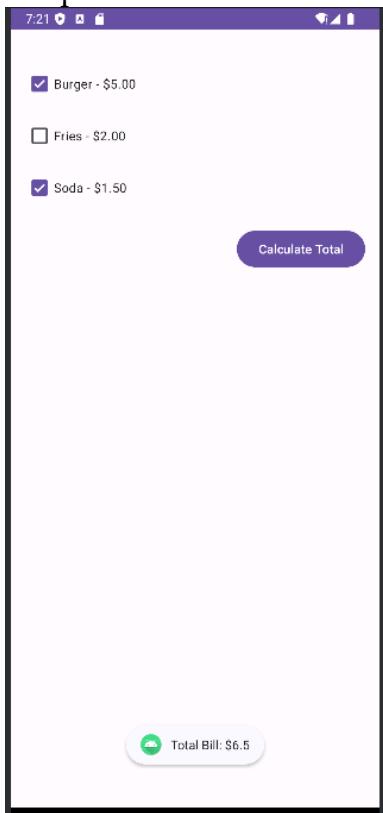
<CheckBox
    android:id="@+id/checkbox_item2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Fries - $2.00"
    android:layout_below="@+id/checkbox_item1"
    android:layout_marginTop="10dp"/>

<CheckBox
    android:id="@+id/checkbox_item3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Soda - $1.50"
    android:layout_below="@+id/checkbox_item2"
    android:layout_marginTop="10dp"/>

<Button
    android:id="@+id/button_calculate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Calculate Total"
    android:layout_below="@+id/checkbox_item3"
    android:layout_marginTop="20dp"
    android:layout_alignParentEnd="true"/>

</RelativeLayout>
```

Output:



8. Write an android application for rating a movie on the scale of 1 to 5. Display the result in TextView.

Solution:

```
package com.example.forpractice;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Find the RadioGroup, Button, and TextView by their IDs
        RadioGroup radioGroupRating = findViewById(R.id.radioGroup_rating);
        Button buttonSubmit = findViewById(R.id.button_submit);
        TextView textViewResult = findViewById(R.id.textView_result);

        // Set an OnClickListener on the Submit Button
        buttonSubmit.setOnClickListener(v -> {
            // Get the selected RadioButton ID
            int selectedId = radioGroupRating.getCheckedRadioButtonId();

            // Find the RadioButton by ID
            RadioButton selectedRadioButton = findViewById(selectedId);

            // Get the text from the selected RadioButton
            String rating = selectedRadioButton != null ?
                selectedRadioButton.getText().toString() : "No rating selected";

            // Display the selected rating in the TextView
            textViewResult.setText("You rated the movie: " + rating);
        });
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <TextView
        android:id="@+id/textView_rating"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Rate the Movie (1 to 5):"
        android:textSize="18sp"
        android:layout_marginBottom="16dp"/>

    <RadioGroup
        android:id="@+id/radioGroup_rating"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView_rating">

    <RadioButton
        android:id="@+id radioButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1"/>

    <RadioButton
        android:id="@+id radioButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="2"/>

    <RadioButton
        android:id="@+id radioButton3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="3"/>

    <RadioButton
        android:id="@+id radioButton4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="4"/>

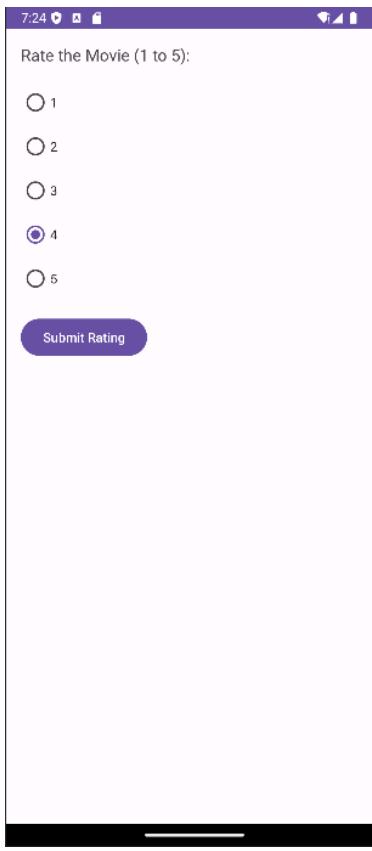
    <RadioButton
        android:id="@+id radioButton5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="5"/>

</RadioGroup>

<Button
    android:id="@+id/button_submit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Submit Rating"
    android:layout_below="@+id/radioGroup_rating"
    android:layout_marginTop="16dp"/>

<TextView
    android:id="@+id/textView_result"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:textSize="18sp"
    android:layout_below="@+id/button_submit"
    android:layout_marginTop="16dp"/>
</RelativeLayout>
```

Output:



9. Write an android application using Spinner for selecting the Cricket Team Players from the given list. Display the name of the Cricketer in Toast.

Solution:

```
package com.example.forpractice;

import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Find the Spinner and Button by their IDs
        Spinner spinnerPlayers = findViewById(R.id.spinner_players);
        Button buttonShow = findViewById(R.id.button_show);

        // Array of players
        String[] players = {"Virat Kohli", "Rohit Sharma", "Steve Smith", "Kane Williamson", "Joe Root"};

        // Create an ArrayAdapter using the string array and a default spinner
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item, players);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinnerPlayers.setAdapter(adapter);

        buttonShow.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String selectedPlayer = spinnerPlayers.getSelectedItem().toString();
                Toast.makeText(getApplicationContext(), selectedPlayer, Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

```

layout
    ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
        android.R.layout.simple_spinner_item, players);

    // Specify the layout to use when the list of choices appears
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

    // Apply the adapter to the spinner
    spinnerPlayers.setAdapter(adapter);

    // Set an OnClickListener on the Show Button
    buttonShow.setOnClickListener(v -> {
        // Get the selected item from the Spinner
        String selectedPlayer =
        spinnerPlayers.getSelectedItem().toString();

        // Display the selected player in a Toast
        Toast.makeText(MainActivity.this, "Selected Player: " +
selectedPlayer, Toast.LENGTH_SHORT).show();
    });
}
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <Spinner
        android:id="@+id/spinner_players"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"/>

    <Button
        android:id="@+id/button_show"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Selected Player"
        android:layout_below="@+id/spinner_players"
        android:layout_marginTop="20dp"/>
</RelativeLayout>

```

Output:



10. Write an android application which will demonstrate the use of ImageButton.

Solution:

```
package com.example.forpractice;

import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Find the ImageButton by its ID
        ImageButton imageView = findViewById(R.id.imageButton);

        // Set an OnClickListener on the ImageButton
        imageView.setOnClickListener(v -> {
            // Display a Toast message when the ImageButton is clicked
            Toast.makeText(MainActivity.this, "ImageButton clicked!", Toast.LENGTH_SHORT).show();
        });
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <ImageButton
        android:id="@+id/imageButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:src="@drawable/viratkohli"
        android:contentDescription="ImageButton"
        android:background="?attr/selectableItemBackgroundBorderless"
        android:layout_marginTop="20dp"/>

</RelativeLayout>
```

Output:





**Pimpri Chinchwad Education Trust's
Pimpri Chinchwad College of Engineering**

Assignment-02

Roll No:123M1H010

Name of Student:Harshal Bhamare

Submission Date: 04 / 09 / 2024

Q.1 Write an android application which will allow users to navigate from one activity to another activity. The first Activity will ask the user to enter the name user and the Second activity will display the name in TextView which was entered in the first activity.

Solution:

xml:

activity_main:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/editTextName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your name" />

    <Button
        android:id="@+id/buttonSubmit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Submit" />

</LinearLayout>
```

activity_second:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/textViewName"
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:text="Your name will be displayed here" />

</LinearLayout>
```

java:

mainactivity:

```
package com.example.la2q1;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        EditText editTextName = findViewById(R.id.editTextName);
        Button buttonSubmit = findViewById(R.id.buttonSubmit);

        buttonSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String userName = editTextName.getText().toString();

                Intent intent = new Intent(MainActivity.this, SecondActivity.class);
                intent.putExtra("USER_NAME", userName);

                startActivity(intent);
            }
        });
    }
}
```

secondactivity:

```
package com.example.la2q1;

import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
```

```
public class SecondActivity extends AppCompatActivity {

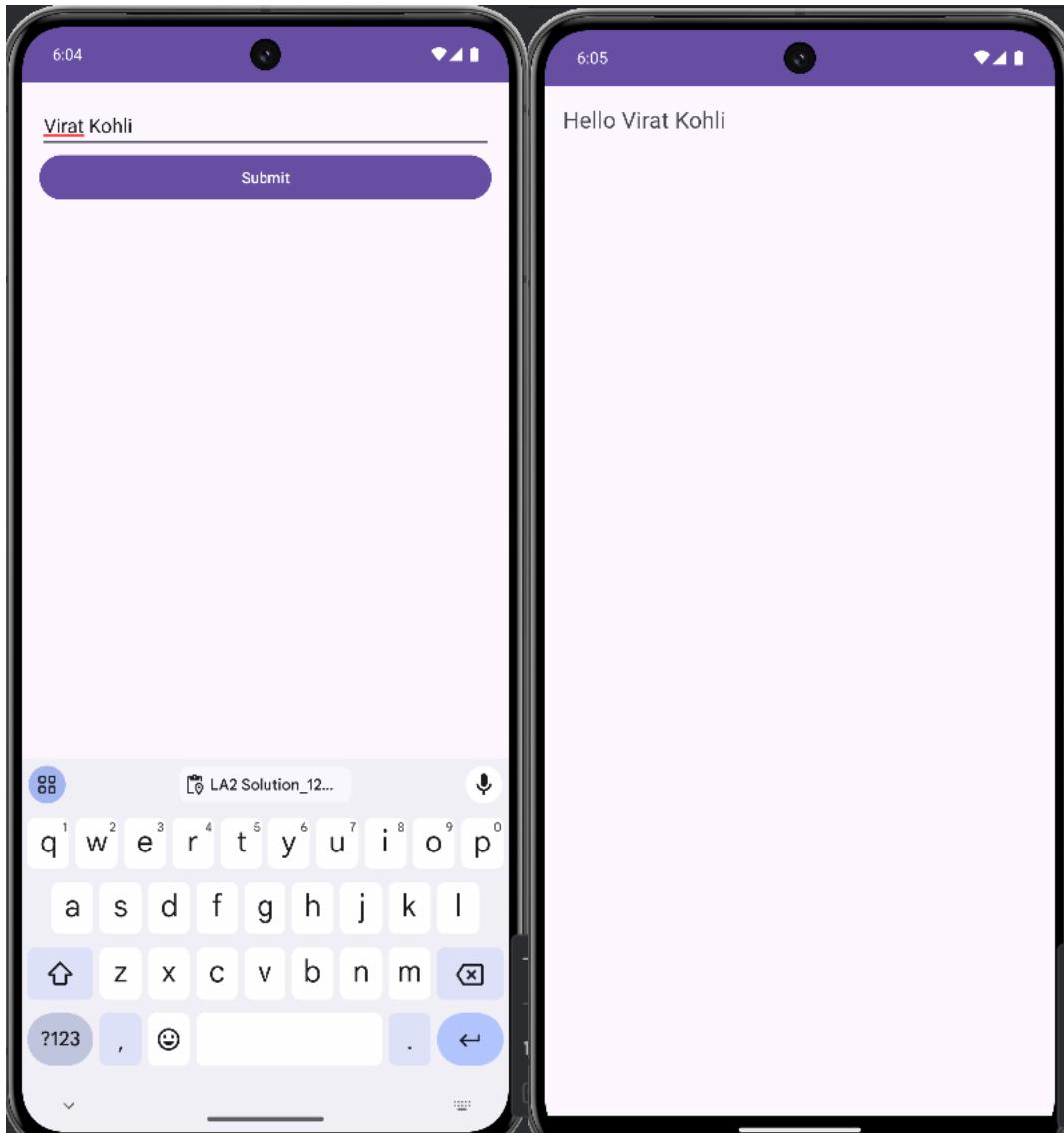
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);

        TextView textViewName = findViewById(R.id.textViewName);

        Intent intent = getIntent();
        String name = intent.getStringExtra("USER_NAME");

        textViewName.setText("Hello " + name);
    }
}
```

Output:



1. Write an android application that asks the user to enter the URL, and after clicking the button, the URL link should be opened in the web browser in an emulator.

Solution:

xml:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

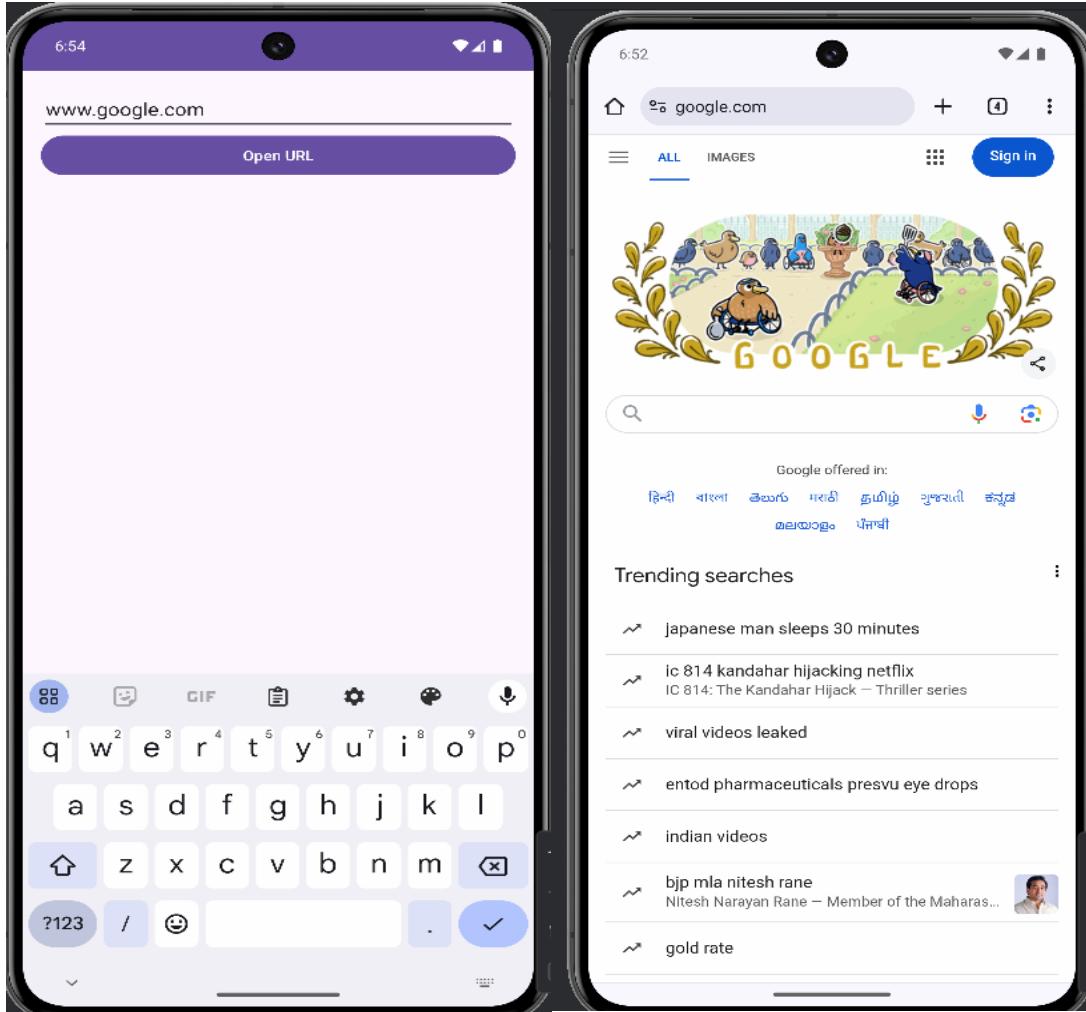
    <EditText
        android:id="@+id/editTextURL"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter URL"
        android:inputType="textUri" />
```

```
<Button  
    android:id="@+id/buttonOpenURL"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Open URL" />  
</LinearLayout>
```

```
java:  
package com.example.la2q2;  
  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        EditText editTextURL = findViewById(R.id.editTextURL);  
        Button buttonOpenURL = findViewById(R.id.buttonOpenURL);  
  
        buttonOpenURL.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                String url = editTextURL.getText().toString().trim();  
  
                if (!url.isEmpty()) {  
                    if (!url.startsWith("http://") && !url.startsWith("https://")) {  
                        url = "http://" + url;  
                    }  
  
                    Intent browserIntent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));  
                    startActivity(browserIntent);  
                } else {  
                    Toast.makeText(MainActivity.this, "Please enter url", Toast.LENGTH_SHORT).show();  
                }  
            }  
        });  
    }  
}
```

```
}
```

Output:



2. Write an android application that will demonstrate the use of BaseAdapter and ArrayAdapter.

Solution:

xml:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    android:padding="16dp">
```

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="ArrayAdapter Example"
    android:textSize="18sp"
    android:paddingBottom="8dp"/>

<ListView
    android:id="@+id/listView ArrayAdapter"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"/>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="BaseAdapter Example"
    android:textSize="18sp"
    android:paddingBottom="8dp"/>

<ListView
    android:id="@+id/listView BaseAdapter"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"/>
</LinearLayout>

```

java:

```

package com.example.la2q3;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.BaseAdapter;
import android.widget.ListView;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ListView listView ArrayAdapter = findViewById(R.id.listView ArrayAdapter);
        String[] fruits = {"Iris", "Marigold", "Rose", "Daisy", "Jasmine", "Lily", "Dahlia"};

        ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1, fruits);

```

```
listViewArrayAdapter.setAdapter(arrayAdapter);

ListView listViewBaseAdapter = findViewById(R.id.listViewBaseAdapter);
CustomAdapter customAdapter = new CustomAdapter(fruits);
listViewBaseAdapter.setAdapter(customAdapter);
}

class CustomAdapter extends BaseAdapter {

    private String[] data;

    public CustomAdapter(String[] data) {
        this.data = data;
    }

    @Override
    public int getCount() {
        return data.length;
    }

    @Override
    public Object getItem(int position) {
        return data[position];
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        ViewHolder holder;

        if (convertView == null) {
            convertView = LayoutInflater.from(parent.getContext()).inflate(android.R.layout.simple_list_item_2,
parent, false);
            holder = new ViewHolder();
            holder.textView1 = convertView.findViewById(android.R.id.text1);
            holder.textView2 = convertView.findViewById(android.R.id.text2);
            convertView.setTag(holder);
        } else {
            holder = (ViewHolder) convertView.getTag();
        }

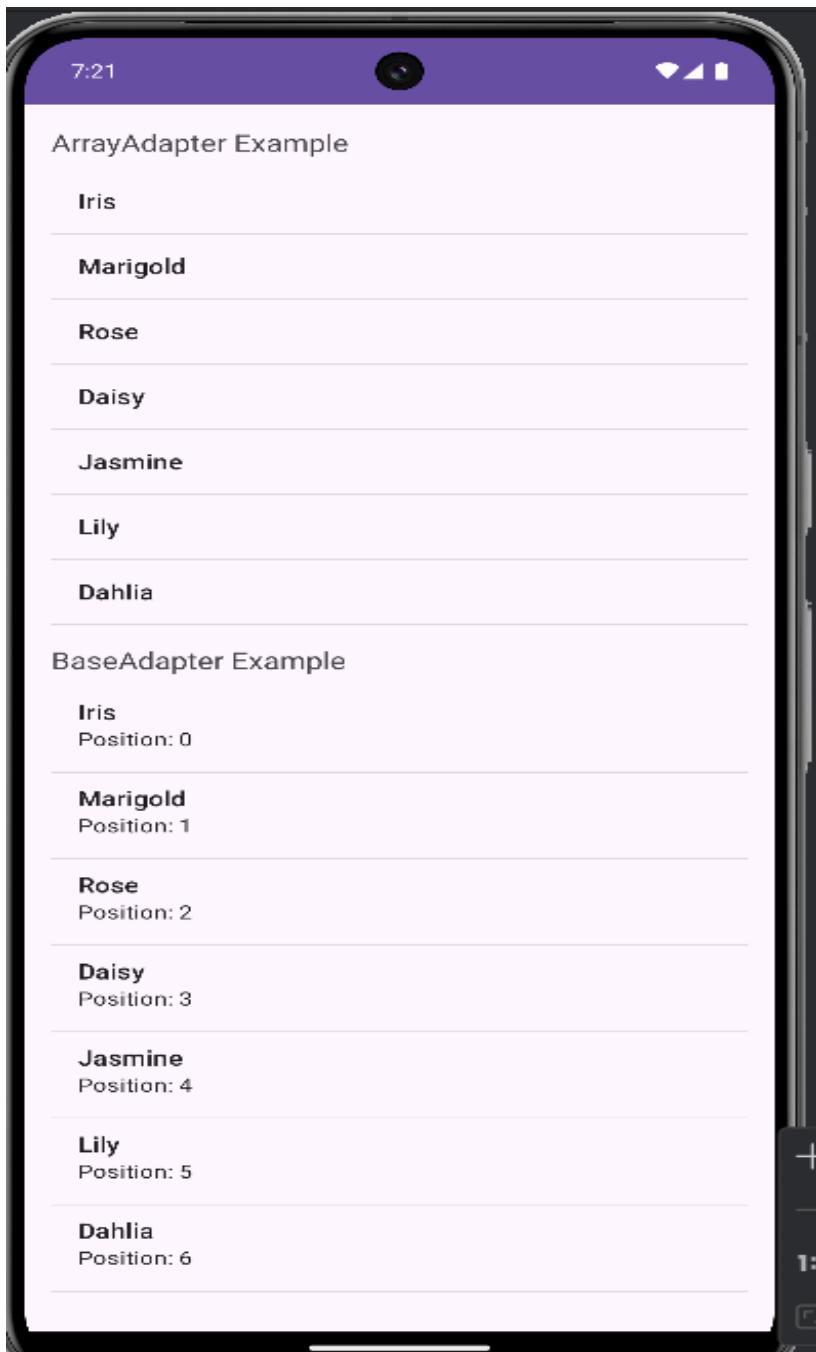
        holder.textView1.setText(data[position]);
        holder.textView2.setText("Position: " + position);

        return convertView;
    }

    class ViewHolder {
        TextView textView1;
```

```
        TextView textView2;  
    }  
}  
}
```

Output:



3. Write an android application for Gallery using adapters.

Solution:

xml:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="8dp">

    <GridView
        android:id="@+id/gridView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:numColumns="3"
        android:verticalSpacing="8dp"
        android:horizontalSpacing="8dp"
        android:stretchMode="columnWidth"/>
</LinearLayout>
```

java:

```
package com.example.la2q4;

import android.content.Context;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.GridView;
import android.widget.ImageView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private Integer[] imageIds = {
        R.drawable.image1, R.drawable.image2, R.drawable.image3,
        R.drawable.image4, R.drawable.image5, R.drawable.image6 };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        GridView gridView = findViewById(R.id.gridView);
        ImageAdapter imageAdapter = new ImageAdapter(this, imageIds);
        gridView.setAdapter(imageAdapter);
    }

    public class ImageAdapter extends BaseAdapter {
        private Context context;
```

```
private Integer[] imageIds;

public ImageAdapter(Context context, Integer[] imageIds) {
    this.context = context;
    this.imageIds = imageIds;
}

@Override
public int getCount() {
    return imageIds.length;
}

@Override
public Object getItem(int position) {
    return imageIds[position];
}

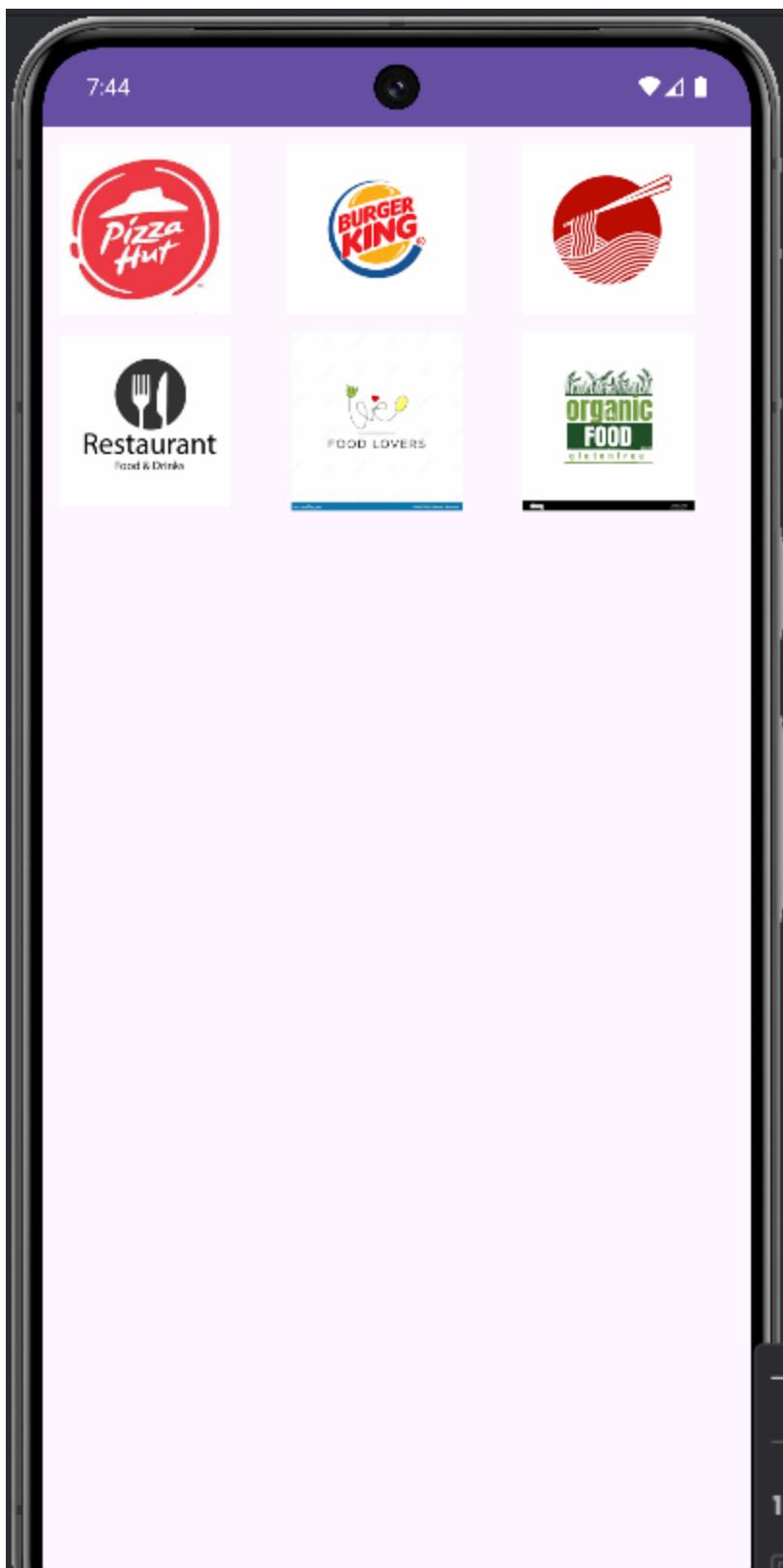
@Override
public long getItemId(int position) {
    return position;
}

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    ImageView imageView;

    if (convertView == null) {
        imageView = new ImageView(context);
        imageView.setLayoutParams(new GridView.LayoutParams(340, 340));
        imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
        imageView.setPadding(8, 8, 8, 8);
    } else {
        imageView = (ImageView) convertView;
    }

    imageView.setImageResource(imageIds[position]);
    return imageView;
}
}
```

Output:



4. Write an application demonstrating the use of Android Session Management.

Solution:

xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        app:title="@string/app_name"
        app:titleTextColor="@color/white"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary" />
    <EditText
        android:id="@+id/idEdtEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/toolbar"
        android:layout_marginStart="10dp"
        android:layout_marginTop="50dp"
        android:layout_marginEnd="10dp"
        android:hint="@string/enter_youe_email"
        android:importantForAutofill="no"
        android:inputType="textEmailAddress" />
    <EditText
        android:id="@+id/idEdtPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/idEdtEmail"
        android:layout_marginStart="10dp"
        android:layout_marginTop="30dp"
        android:layout_marginEnd="10dp"
        android:hint="@string/enter_password"
        android:importantForAutofill="no"
        android:inputType="textPassword" />
    <Button
        android:id="@+id/idBtnLogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/idEdtPassword"
        android:layout_marginStart="10dp"
        android:layout_marginTop="30dp"
        android:layout_marginEnd="10dp"
        android:text="@string/login" />
</RelativeLayout>
```

activity_home.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".HomeActivity">

    <TextView
        android:id="@+id/idTVWelcome"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:padding="5dp"
        android:textAlignment="center"
        android:textSize="20sp" />
    <Button
        android:id="@+id/idBtnLogout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/idTVWelcome"
        android:layout_marginStart="20dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="20dp"
        android:text="@string/logout" />
</RelativeLayout>

```

java:

```

package com.example.la2q5;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    public static final String SHARED_PREFS = "shared_prefs";
    public static final String EMAIL_KEY = "email_key";
    public static final String PASSWORD_KEY = "password_key";
    SharedPreferences sharedpreferences;
    String email, password;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        EditText emailEdt = findViewById(R.id.idEditTextEmail);

```

```
EditText passwordEdt = findViewById(R.id.idEdtPassword);
Button loginBtn = findViewById(R.id.idBtnLogin);
sharedpreferences = getSharedPreferences(SHARED_PREFS,
    Context.MODE_PRIVATE);
email = sharedpreferences.getString(EMAIL_KEY, null);
password = sharedpreferences.getString(PASSWORD_KEY, null);
if (email != null && password != null) {
    Intent i = new Intent(MainActivity.this, HomeActivity.class);
    startActivity(i);
    finish();
}
loginBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (TextUtils.isEmpty(emailEdt.getText().toString()) &&
            TextUtils.isEmpty(passwordEdt.getText().toString())) {
            Toast.makeText(MainActivity.this, "Please Enter Email and Password",
                Toast.LENGTH_SHORT).show();
        } else {
            SharedPreferences.Editor editor = sharedpreferences.edit();

            editor.putString(EMAIL_KEY, emailEdt.getText().toString());
            editor.putString(PASSWORD_KEY, passwordEdt.getText().toString());
            editor.apply();
            Intent i = new Intent(MainActivity.this, HomeActivity.class);
            startActivity(i);
            finish();
        }
    }
});
```

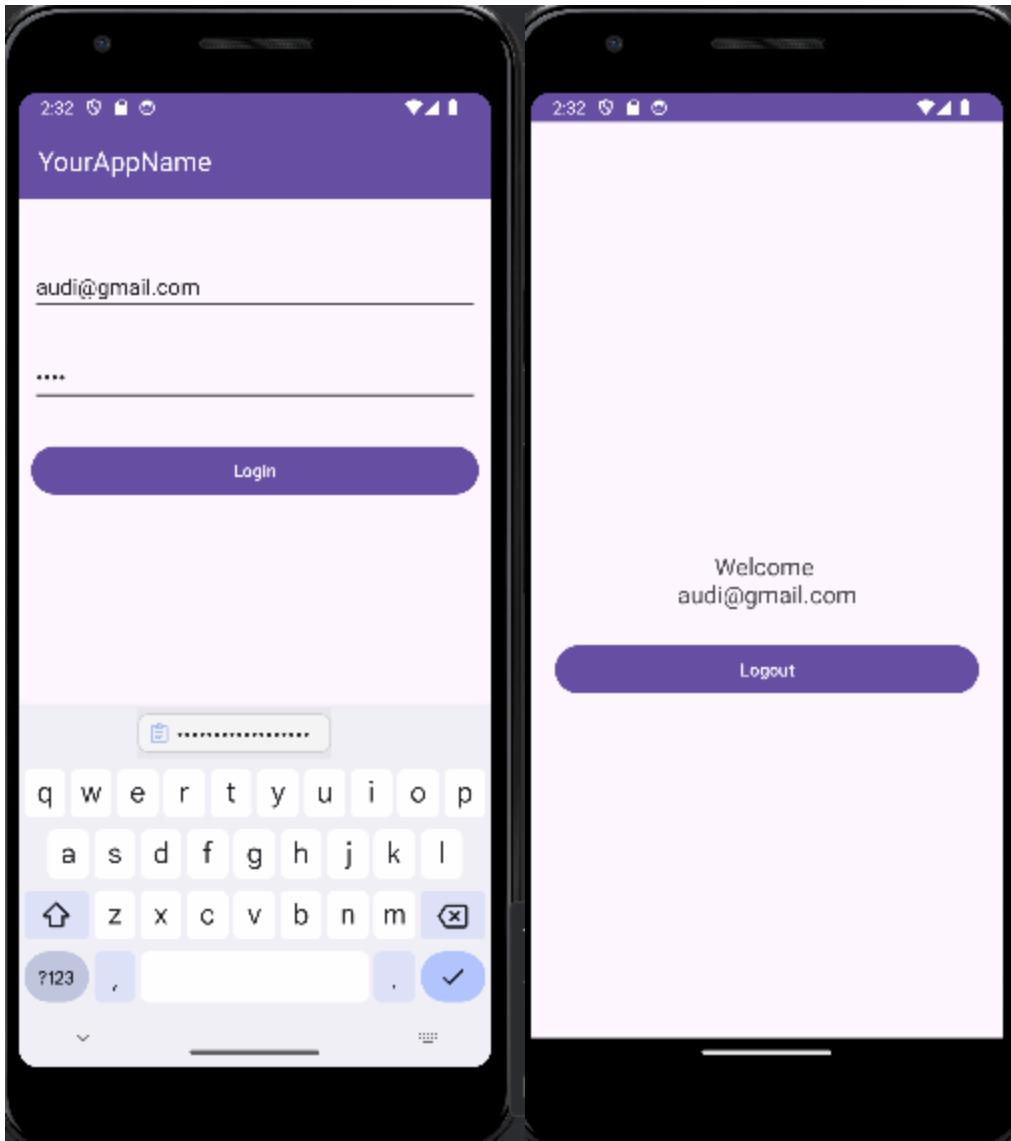
HomeActivity.java:

```
package com.example.la2q5;
import android.content.Context;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
public class HomeActivity extends AppCompatActivity {
    public static final String SHARED_PREFS = "shared_prefs";
    public static final String EMAIL_KEY = "email_key";
    public static final String PASSWORD_KEY = "password_key";
    SharedPreferences sharedpreferences;
    String email;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_home);
sharedpreferences = getSharedPreferences(SHARED_PREFS,
    Context.MODE_PRIVATE);
email = sharedpreferences.getString(EMAIL_KEY, null);
TextView welcomeTV = findViewById(R.id.idTVWelcome);
welcomeTV.setText("Welcome \n" + email);
Button logoutBtn = findViewById(R.id.idBtnLogout);
logoutBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SharedPreferences.Editor editor = sharedpreferences.edit();
        editor.clear();
        editor.apply();
        Intent i = new Intent(HomeActivity.this, MainActivity.class);
        startActivity(i);
        finish();
    }
});
```

Output:



5. Write an android application which will create three fragments in a single activity.

Solution:

xml:

activitymain:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <FrameLayout
        android:id="@+id/fragmentContainer1"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#FFCDD2"/>
```

```
<FrameLayout
    android:id="@+id/fragmentContainer2"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:background="#C8E6C9"/>

<FrameLayout
    android:id="@+id/fragmentContainer3"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:background="#BBDEFB"/>
</LinearLayout>
```

fragment_one:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fragment One"
        android:textSize="24sp"
        android:textColor="#000000"/>
</LinearLayout>
```

fragment_two:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fragment Two"
        android:textSize="24sp"
        android:textColor="#000000"/>
</LinearLayout>
```

fragment_three:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fragment Three"
        android:textSize="24sp"
        android:textColor="#000000"/>
</LinearLayout>
```

java:

mainactivity:

```
package com.example.la2q6;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        loadFragment(new fragment_one(), R.id.fragmentContainer1);
        loadFragment(new fragment_two(), R.id.fragmentContainer2);
        loadFragment(new fragment_three(), R.id.fragmentContainer3);
    }

    private void loadFragment(Fragment fragment, int containerId) {
        FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
        transaction.replace(containerId, fragment);
        transaction.commit();
    }
}
```

fragment_one:

```
package com.example.la2q6;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
```

```
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class fragment_one extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_one, container, false);
    }
}
```

fragment_two:

```
// src/java/com/example/yourapp/FragmentTwo.java
package com.example.la2q6;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class fragment_two extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_two, container, false);
    }
}
```

fragment_three:

```
package com.example.la2q6;

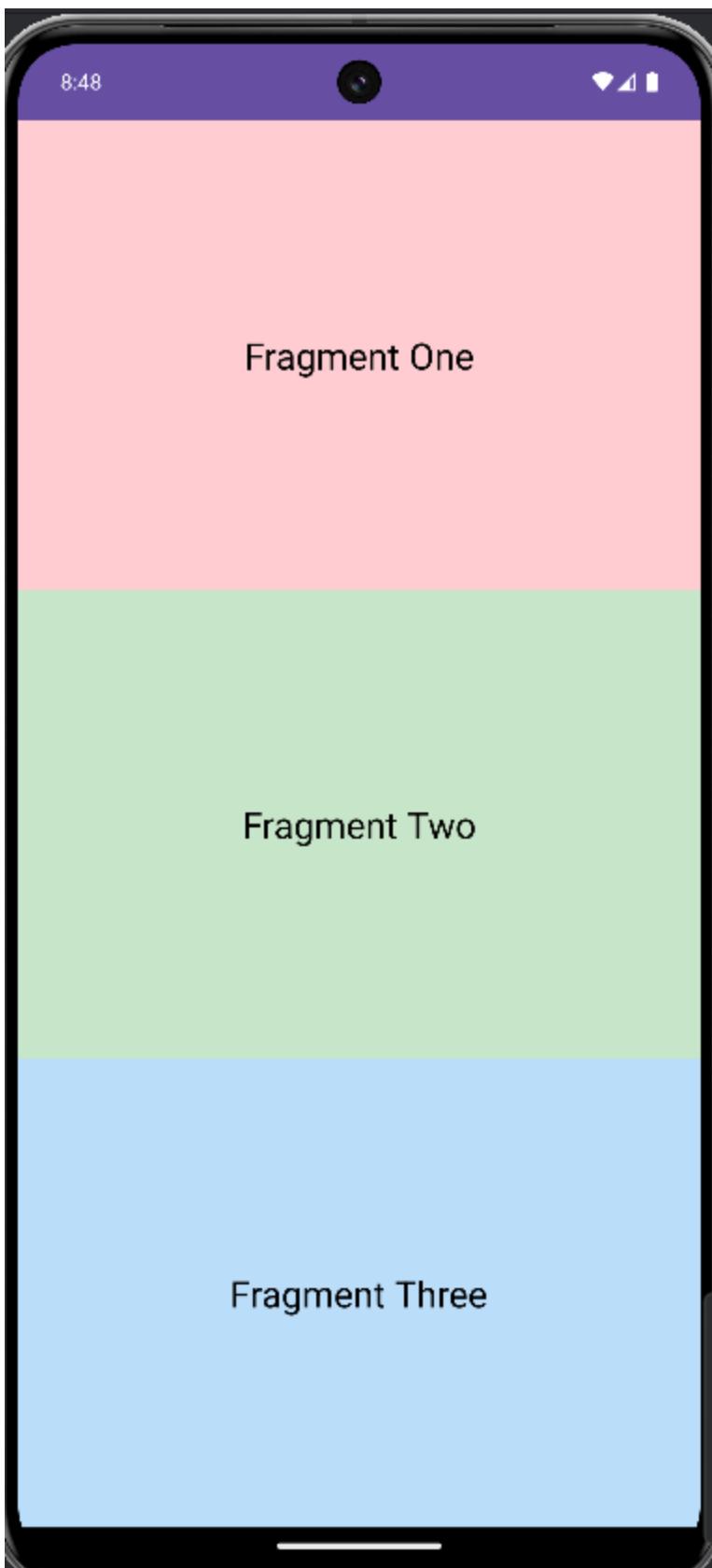
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class fragment_three extends Fragment {

    @Nullable
    @Override
```

```
public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
    return inflater.inflate(R.layout.fragment_three, container, false);
}
```

Output:



6. Write an android application for Fragment Activity Life Cycle.

Solution:

xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">
    <FrameLayout
        android:id="@+id/fragmentContainer"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="@android:color/darker_gray" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center">
        <Button
            android:id="@+id/buttonAddFragment"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

            android:text="Add Fragment" />
        <Button
            android:id="@+id/buttonRemoveFragment"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Remove Fragment"
            android:layout_marginStart="16dp" />
    </LinearLayout>
</LinearLayout>
```

activity_fragment_lifecycle.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center">
    <TextView
        android:id="@+id/lifecycleTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Fragment Lifecycle States:"
        android:textSize="18sp" />
</LinearLayout>
```

java:

```
package com.example.la2q7;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentTransaction;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
    private Button buttonAddFragment, buttonRemoveFragment;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        buttonAddFragment = findViewById(R.id.buttonAddFragment);
        buttonRemoveFragment = findViewById(R.id.buttonRemoveFragment);
        buttonAddFragment.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                addFragment();
            }
        });
        buttonRemoveFragment.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                removeFragment();
            }
        });
    }
    private void addFragment() {
        FragmentLifecycleDemo fragment = new FragmentLifecycleDemo();

        FragmentTransaction transaction =
            getSupportFragmentManager().beginTransaction();
        transaction.add(R.id.fragmentContainer, fragment, "LIFECYCLE_FRAGMENT");
        transaction.addToBackStack(null); // To handle back navigation
        transaction.commit();
    }
    private void removeFragment() {
        Fragment fragment =
            getSupportFragmentManager().findFragmentByTag("LIFECYCLE_FRAGMENT");
        if (fragment != null) {
            FragmentTransaction transaction =
                getSupportFragmentManager().beginTransaction();
            transaction.remove(fragment);
            transaction.commit();
        }
    }
}
```

fragmentLifecycle.java:

```
package com.example.la2q7;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
public class FragmentLifecycleDemo extends Fragment {
    private static final String TAG = "FragmentLifecycle";
    private TextView lifecycleTextView;
    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.d(TAG, "onCreate called");
    }
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup
            container,
            @Nullable Bundle savedInstanceState) {
        Log.d(TAG, "onCreateView called");
        View view = inflater.inflate(R.layout.activity_fragment.lifecycle_demo, container, false);

        lifecycleTextView = view.findViewById(R.id.lifecycleTextView);
        updateLifecycleState("onCreateView");
        return view;
    }
    @Override
    public void onStart() {
        super.onStart();
        Log.d(TAG, "onStart called");
        updateLifecycleState("onStart");
    }
    @Override
    public void onResume() {
        super.onResume();
        Log.d(TAG, "onResume called");
        updateLifecycleState("onResume");
    }
    @Override
    public void onPause() {
        super.onPause();
        Log.d(TAG, "onPause called");
        updateLifecycleState("onPause");
    }
    @Override
    public void onStop() {
        super.onStop();
```

```
        Log.d(TAG, "onStop called");
        updateLifecycleState("onStop");
    }

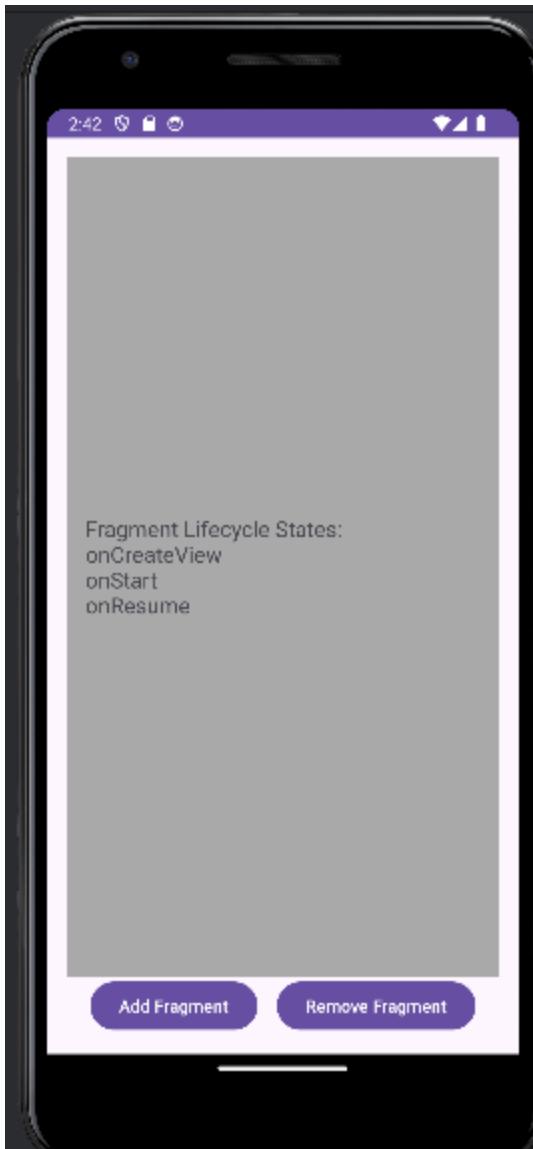
    @Override
    public void onDestroyView() {
        super.onDestroyView();
        Log.d(TAG, "onDestroyView called");
        updateLifecycleState("onDestroyView");
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        Log.d(TAG, "onDestroy called");

        updateLifecycleState("onDestroy");
    }

    private void updateLifecycleState(String state) {
        if (lifecycleTextView != null) {
            String currentText = lifecycleTextView.getText().toString();
            lifecycleTextView.setText(currentText + "\n" + state);
        }
    }
}
```

Output:



7. Write an android application that will look like WhatsApp Application using Fragment.

Solution:

xml:

Fraagement_contacts.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- Contacts Fragment Layout -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Whatsapp contacts"
```

```
    android:textSize="24sp"/>
</RelativeLayout>
```

Fragment_settings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- Settings Fragment Layout -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Whatsapp settings"
        android:textSize="24sp"/>
</RelativeLayout>
```

Fragment_chat.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- Settings Fragment Layout -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Whatsapp settings"
        android:textSize="24sp"/>
</RelativeLayout>
```

Activitymain.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/fragment_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="fill"/>

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_navigation"
        android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"
    app:menu="@menu/bottom_nav_menu"
    app:labelVisibilityMode="labeled"/>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

java:

```
package com.example.myapplication;

import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;
import com.google.android.material.bottomnavigation.BottomNavigationView;

public class MainActivity extends AppCompatActivity {

    private BottomNavigationView bottomNavigationView;
    private FragmentManager fragmentManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        bottomNavigationView = findViewById(R.id.bottom_navigation);
        fragmentManager = getSupportFragmentManager();

        bottomNavigationView.setOnNavigationItemSelected(item -> {
            Fragment selectedFragment = null;

            if (item.getItemId() == R.id.nav_chats) {
                selectedFragment = new ChatFragment();
            } else if (item.getItemId() == R.id.nav_contacts) {
                selectedFragment = new ContactsFragment();
            } else if (item.getItemId() == R.id.nav_settings) {
                selectedFragment = new SettingsFragment();
            }

            if (selectedFragment != null) {
                FragmentTransaction transaction = fragmentManager.beginTransaction();
                transaction.replace(R.id.fragment_container, selectedFragment);
                transaction.commit();
            }

            return true;
        });
    }
}
```

```
// Set default fragment  
    bottomNavigationView.setSelectedItemId(R.id.nav_chats);  
}  
}
```

Settingsfragment.java:

```
package com.example.myapplication;  
  
import android.os.Bundle;  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
public class SettingsFragment extends Fragment {  
  
    @Nullable  
    @Override  
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
        return inflater.inflate(R.layout.fragment_settings, container, false);  
    }  
}
```

Contactfragemtns.java:

```
package com.example.myapplication;  
  
import android.os.Bundle;  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
public class ContactsFragment extends Fragment {  
  
    @Nullable  
    @Override  
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
        return inflater.inflate(R.layout.fragment_contacts, container, false);  
    }  
}
```

Chatfragment.java:

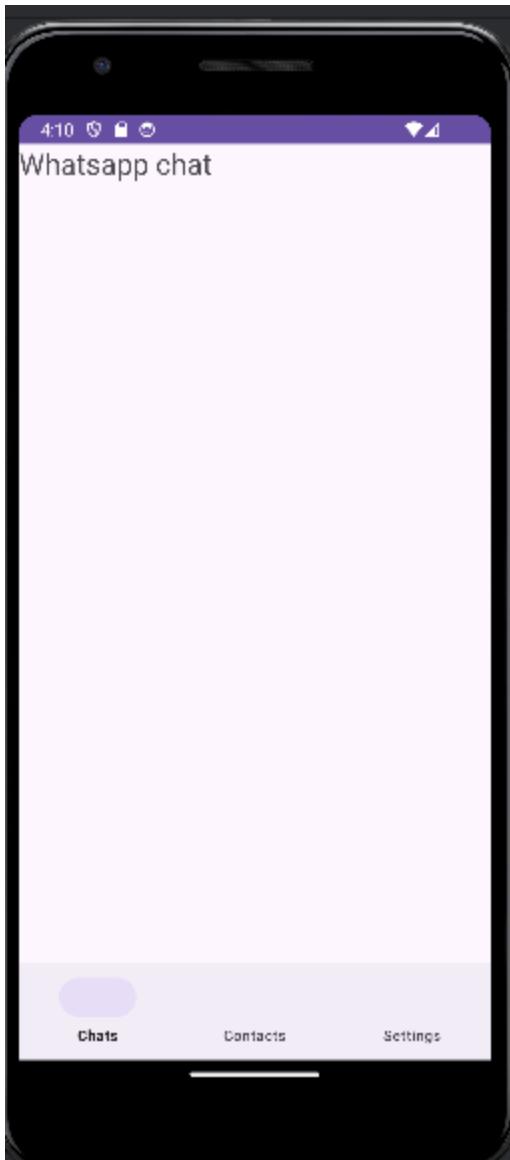
```
package com.example.myapplication;  
  
import android.os.Bundle;
```

```
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class ChatFragment extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_chat, container, false);
    }
}
```

Output:



8. Write an android application that will parse XML data

Solution:

xml file:

```
<?xml version="1.0"?>
<records>
<car>
<name>Audi</name>
<price>5000000</price>
</car>
<car>
<name>BMW</name>
<price>7000000</price>
</car>
<car>
<name>Mercedes-Benz</name>
```

```
<price>10000000</price>
</car>
</records>
```

xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tv1"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

java:

```
package com.example.la2q9;
import java.io.InputStream;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends Activity {
    TextView tv1;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tv1=(TextView)findViewById(R.id.tv1);
        try {
```

```
InputStream is = getAssets().open("new.xml");

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
Document doc = dBuilder.parse(is);

Element element=doc.getDocumentElement();
element.normalize();

NodeList nList = doc.getElementsByTagName("car");
for (int i=0; i<nList.getLength(); i++) {

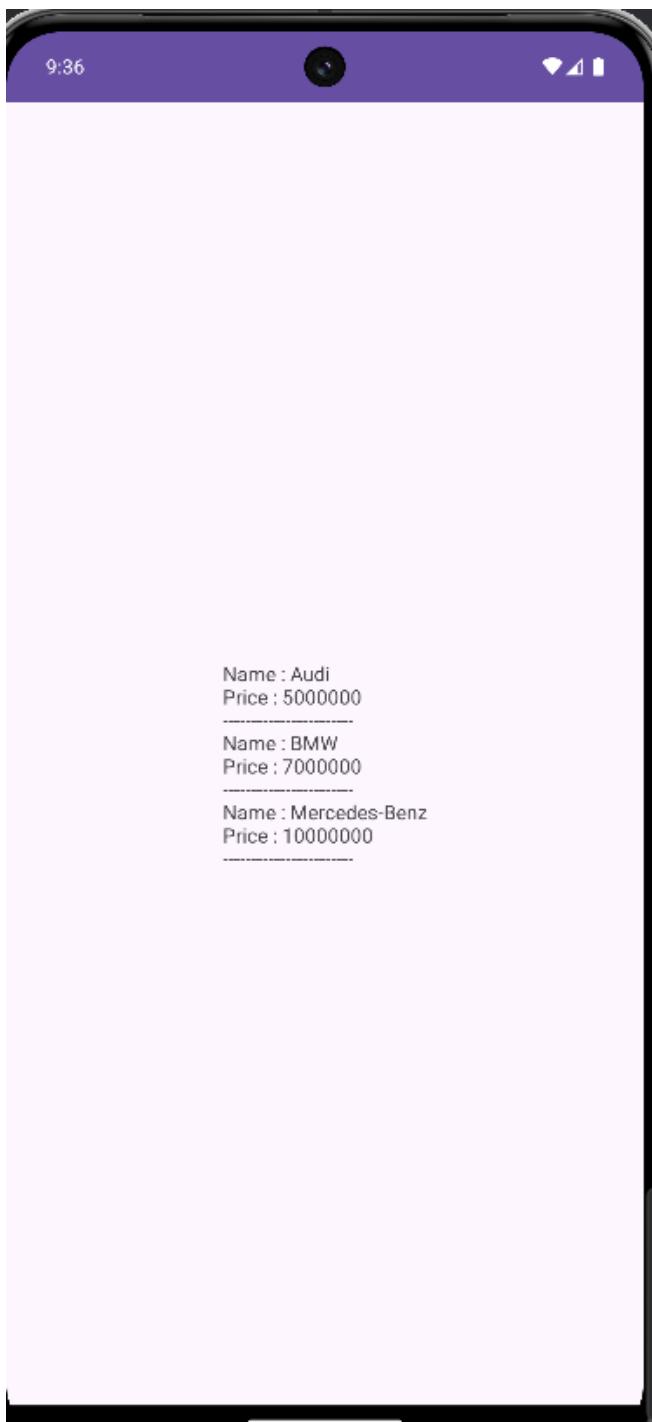
    Node node = nList.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element2 = (Element) node;
        tv1.setText(tv1.getText()+"\nName : " + getValue("name", element2)+"\n");
        tv1.setText(tv1.getText()+"Price : " + getValue("price", element2)+"\n");
        tv1.setText(tv1.getText()+"-----");
    }
}

} catch (Exception e) {e.printStackTrace();}

}

private static String getValue(String tag, Element element) {
    NodeList nodeList = element.getElementsByTagName(tag).item(0).getChildNodes();
    Node node = (Node) nodeList.item(0);
    return node.getNodeValue();
}
}
```

Output:



9. Write an android application that will parse JSON data

Solution:

Json file:

```
[  
  {  
    "name": "Mango",  
    "price": "500"
```

```
},
{
  "name": "Kiwi",
  "price": "200"
}
]
```

xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tvData"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="75dp"
        android:layout_marginTop="46dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

java:

```
package com.example.la2q10;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.InputStream;
import java.io.IOException;

public class MainActivity extends Activity {

    private TextView tvData;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tvData = findViewById(R.id.tvData);
```

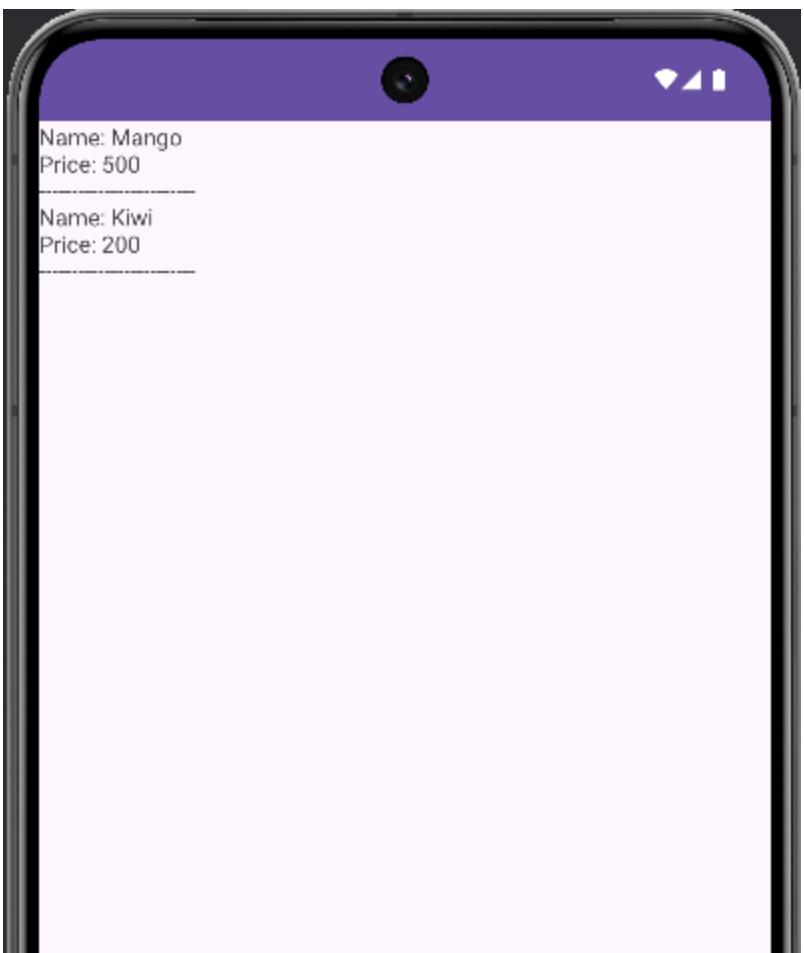
```
try {
    // Load JSON data from assets
    InputStream is = getAssets().open("New.json");
    int size = is.available();
    byte[] buffer = new byte[size];
    is.read(buffer);
    is.close();

    String jsonString = new String(buffer, "UTF-8");
    JSONArray jsonArray = new JSONArray(jsonString);
    StringBuilder sb = new StringBuilder();
    for (int i = 0; i < jsonArray.length(); i++) {
        JSONObject jsonObject = jsonArray.getJSONObject(i);
        String name = jsonObject.getString("name");
        String price = jsonObject.getString("price");
        sb.append("Name: ").append(name).append("\n");
        sb.append("Price: ").append(price).append("\n");
        sb.append("-----\n");
    }

    tvData.setText(sb.toString());
}

} catch (IOException | JSONException e) {
    e.printStackTrace();
}
}
```

Output:





Assignment-03

Roll No:123M1H010

Name of Student: Harshal Bhamare

Submission Date: 27 /09 /2024

1. Design a simple mobile application that displays user profile information. The profile includes a profile picture, user name, and a short bio. The UI should use different ViewGroups (LinearLayout, RelativeLayout, FrameLayout) to organize the layout. Experiment with different layout structures and ensure the profile image and text are aligned properly within the layout.

Solution:

Xml code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginBottom="16dp">

        <ImageView
            android:id="@+id/profile_image"
            android:layout_width="120dp"
            android:layout_height="120dp"
            android:layout_gravity="center"
            android:src="@drawable/profile1"
            android:contentDescription="Profile Picture"
            android:scaleType="centerCrop"
            android:background="@drawable/profile1"/>

    </FrameLayout>
```

```
<RelativeLayout
    android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:padding="16dp">>

    <TextView
        android:id="@+id/user_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tom Cruise "
        android:textSize="24sp"
        android:textStyle="bold"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:paddingBottom="8dp" />

    <TextView
        android:id="@+id/user_bio"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="A short bio "
        android:textSize="16sp"
        android:layout_below="@+id/user_name"
        android:layout_centerHorizontal="true"
        android:textColor="#666"
        android:paddingTop="8dp" />

</RelativeLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:paddingTop="24dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Contact Info:" android:textSize="18sp"
        android:textStyle="bold" android:paddingBottom="8dp"
        />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Email: tom@gmail.com"
        android:textSize="16sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Phone: 1234567890"
        android:textSize="16sp" />

</LinearLayout>
```

</LinearLayout>

Java code:

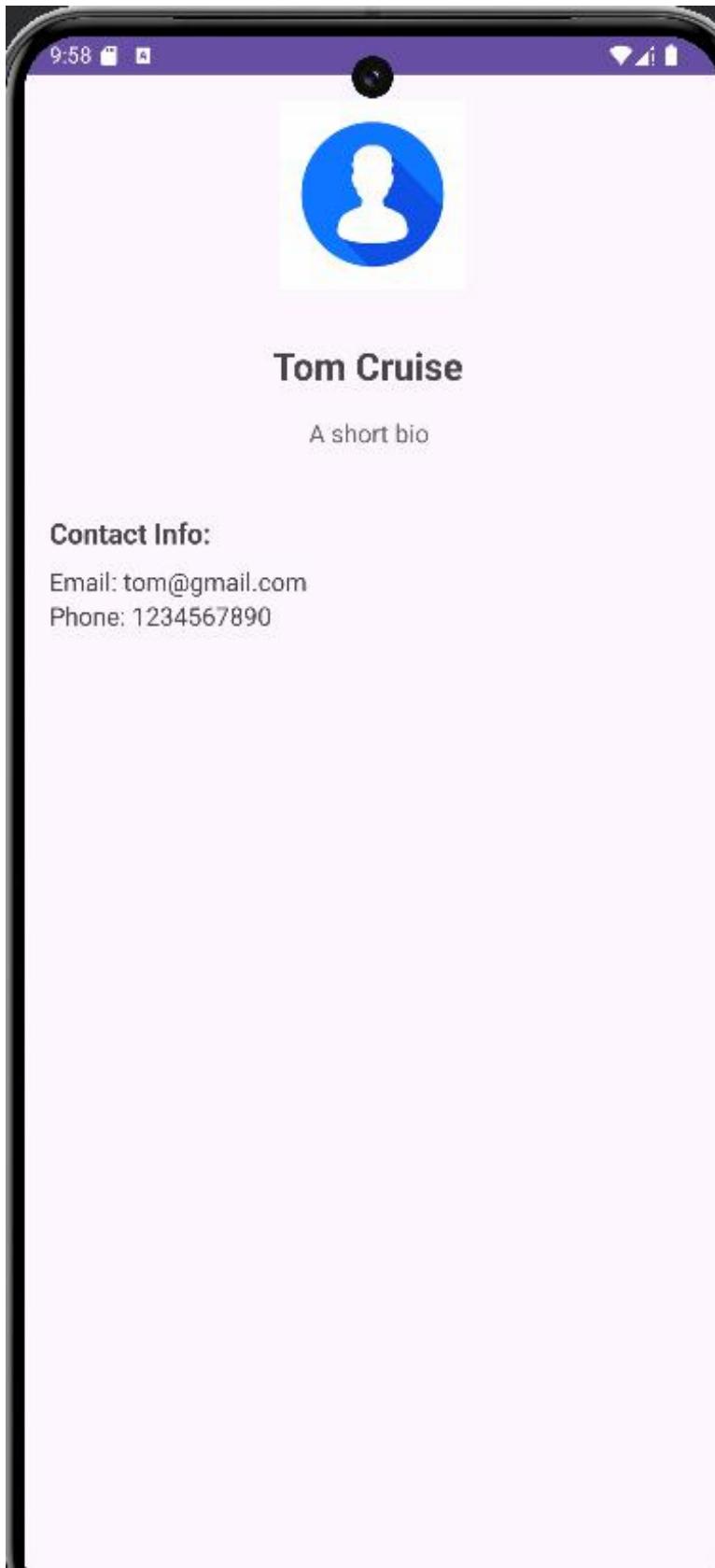
```
package com.example.la3q1;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Output:



2. Develop a user registration form for a mobile application. The form should collect the user's name, email, gender, favorite activities (using checkboxes), and country of residence (using a dropdown Spinner). Once the user fills in the details and clicks the "Submit" button, display the entered information in a confirmation TextView.

Solution:

Xml code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/nameEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name"
        android:inputType="text" />

    <EditText
        android:id="@+id/emailEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress" />

    <RadioGroup
        android:id="@+id/genderRadioGroup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <RadioButton
            android:id="@+id/maleRadioButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
        android:text="Male" />

    <RadioButton
        android:id="@+id/femaleRadioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Female" />

    <RadioButton
        android:id="@+id/otherRadioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Other" />
</RadioGroup>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <CheckBox
        android:id="@+id/sportsCheckBox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sports" />

    <CheckBox
        android:id="@+id/musicCheckBox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Music" />

    <CheckBox
        android:id="@+id/travelCheckBox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Travel" />
</LinearLayout>

<Spinner
    android:id="@+id/countrySpinner"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<Button
    android:id="@+id/submitButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Submit" />

<TextView
    android:id="@+id/confirmationTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingTop="16dp"
```

```
    android:textSize="16sp" />
</LinearLayout>
```

Java code:

```
package com.example.la3q2;

import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        final EditText nameEditText = findViewById(R.id.nameEditText); final EditText emailEditText =
        findViewById(R.id.emailEditText); final RadioGroup genderRadioGroup =
        findViewById(R.id.genderRadioGroup);
        final CheckBox sportsCheckBox = findViewById(R.id.sportsCheckBox); final CheckBox musicCheckBox =
        findViewById(R.id.musicCheckBox); final CheckBox travelCheckBox =
        findViewById(R.id.travelCheckBox); final Spinner countrySpinner = findViewById(R.id.countrySpinner);
        final Button submitButton = findViewById(R.id.submitButton); final TextView confirmationTextView =
        findViewById(R.id.confirmationTextView);

        String[] countries = {"Select Country", "USA", "Canada", "UK", "Germany", "France"};
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
        android.R.layout.simple_spinner_item, countries);

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item) ;

        countrySpinner.setAdapter(adapter);

        submitButton.setOnClickListener(new View.OnClickListener() { @Override
            public void onClick(View v) {
                String name = nameEditText.getText().toString(); String email =
                emailEditText.getText().toString(); int selectedGenderId =
                genderRadioGroup.getCheckedRadioButtonId(); RadioButton
                selectedGenderButton =
                findViewById(selectedGenderId);
                String gender = selectedGenderButton != null ?
                selectedGenderButton.getText().toString() : "Not Specified";
            }
        });
    }
}
```

```
        StringBuilder favoriteActivities = new StringBuilder(); if  
            (sportsCheckBox.isChecked())  
favoriteActivities.append("Sports, ");  
                if (musicCheckBox.isChecked())  
favoriteActivities.append("Music, ");  
                    if (travelCheckBox.isChecked())  
favoriteActivities.append("Travel, ");  
                        if (favoriteActivities.length() > 0) { favoriteActivities.setLength(favoriteActivities.length() -  
2);  
}  
String country = countrySpinner.getSelectedItem().toString();  
  
String confirmationText = String.format(  
                "Name: %s\nEmail: %s\nGender: %s\nFavorite Activities:  
%s\nCountry: %s",  
                    name, email, gender, favoriteActivities.toString(),  
country  
);  
  
confirmationTextView.setText(confirmationText);  
    }  
};  
}  
}
```

Output:

10:34

Tom Cruise

tom@gmail.comS

Male Female Other

Sports

Music

Travel

USA

Submit

Name: Tom Cruise
Email: tom@gmail.com
Gender: Male
Favorite Activities: Sports, Music, Travel
Country: USA

3. Create an application that displays a list of countries and their corresponding flags using a RecyclerView. Each list item should be custom-designed to include a country's name and its flag. The user should be able to scroll through the list, and clicking on any country name should display a toast with the selected country name.

Solution:

Xml code:

activity_main:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scrollbars="vertical" />

</LinearLayout>
```

item_country:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="8dp">

    <ImageView
        android:id="@+id/countryFlag"
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:layout_marginEnd="16dp"
        android:contentDescription="Country Flag" />

    <TextView
        android:id="@+id/countryName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="30sp"
        android:textColor="#000" />

</LinearLayout>
```

Java code:

MainActivity:

```
package com.example.la3q3;

import android.os.Bundle;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    RecyclerView recyclerView;
    CountryAdapter adapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        recyclerView = findViewById(R.id.recyclerView);

        List<Country> countryList = getCountryList();

        adapter = new CountryAdapter(countryList, countryName -> {
            Toast.makeText(MainActivity.this, "Selected Country: " +
countryName, Toast.LENGTH_SHORT).show(); });
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        recyclerView.setAdapter(adapter);
    }

    private List<Country> getCountryList() { List<Country> countryList = new ArrayList<>();
        countryList.add(new Country("India", R.drawable.india)); countryList.add(new
        Country("United States", R.drawable.usa)); countryList.add(new Country("Brazil",
        R.drawable.brazil)); countryList.add(new Country("Canada", R.drawable.canada));
        countryList.add(new Country("Australia", R.drawable.australia));

        countryList.add(new Country("United Kingdom",
        R.drawable.unitedkingdom));
        countryList.add(new Country("Mexico ", R.drawable.mexico)); countryList.add(new Country("South
        Africa", R.drawable.southafrica)); countryList.add(new Country("Argentina ", R.drawable.argentina));
        countryList.add(new Country("New Zealand", R.drawable.newzealand));

        countryList.add(new Country("Spain ", R.drawable.spain)); countryList.add(new Country("Italy ",
        R.drawable.italy)); countryList.add(new Country("South Korea", R.drawable.southkorea));
        countryList.add(new Country("Sweden ", R.drawable.sweden)); countryList.add(new
        Country("Netherlands ", R.drawable.netherlands)); return countryList;
    }
}
```

```
}
```

Country:

```
package com.example.la3q3;

public class Country {
    private String name;
    private int flagResId;

    public Country(String name, int flagResId) { this.name = name;
        this.flagResId = flagResId;
    }

    public String getName() {
        return name;
    }

    public int getFlagResId() {
        return flagResId;
    }
}
```

CountryAdapter:

```
package com.example.la3q3;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView; import java.util.List;

public class CountryAdapter extends
RecyclerView.Adapter<CountryAdapter.CountryViewHolder> {

    private List<Country> countryList; private
    OnCountryClickListener listener;

    public CountryAdapter(List<Country> countryList, OnCountryClickListener listener) {
        this.countryList = countryList;
        this.listener = listener;
    }

    @NonNull
    @Override
    public CountryViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
```

```
View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_country, parent,
false);
    return new CountryViewHolder(view);
}

@Override
public void onBindViewHolder(@NonNull CountryViewHolder holder, int position) {
    Country country = countryList.get(position);
    holder.countryName.setText(country.getName());
    holder.countryFlag.setImageResource(country.getFlagResId());

    holder.itemView.setOnClickListener(v ->
listener.onCountryClick(country.getName()));
}

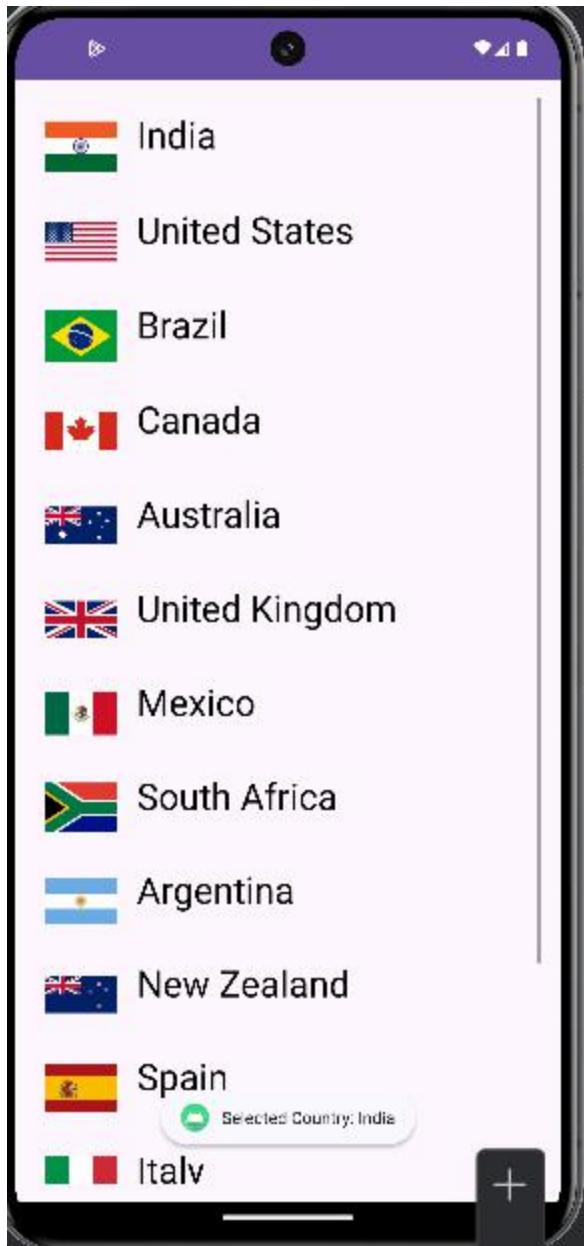
@Override
public int getItemCount() {
    return countryList.size();
}

public static class CountryViewHolder extends RecyclerView.ViewHolder { TextView countryName;
ImageView countryFlag;

public CountryViewHolder(@NonNull View itemView) {
    super(itemView);
    countryName = itemView.findViewById(R.id.countryName);    countryFlag = itemView.findViewById(R.id.countryFlag);
}
}

public interface OnCountryClickListener { void
onCountryClick(String countryName);
}
}
```

Output:



4. Design an event scheduling application where the user can select the date and time of an event. The app should allow the user to pick a date using a DatePickerDialog and a time using a TimePickerDialog. Once the user selects the date and time, display the event details (including the selected date and time) in a TextView.

Solution:

Xml code:

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/dateButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Select Date"
        android:layout_gravity="center_horizontal"
        android:padding="16dp" />

    <Button
        android:id="@+id/timeButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Select Time"
        android:layout_gravity="center_horizontal"
        android:padding="16dp" />

    <Button
        android:id="@+id/saveButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Save Event"
        android:layout_gravity="center_horizontal"
        android:padding="16dp" />

    <TextView
        android:id="@+id/eventDetailsText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:paddingTop="24dp"
        android:text="Event details will be displayed here"
        android:textSize="18sp" />

</LinearLayout>

```

Java code:

```

package com.example.la3q4;

import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

```

```
import androidx.appcompat.app.AppCompatActivity; import  
java.util.Calendar;  
  
public class MainActivity extends AppCompatActivity {
```

```
Button dateButton, timeButton, saveButton; TextView
eventDetailsText;
int selectedYear, selectedMonth, selectedDay, selectedHour, selectedMinute;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    dateButton = findViewById(R.id.dateButton); timeButton =
    findViewById(R.id.timeButton); saveButton = findViewById(R.id.saveButton);
    eventDetailsText = findViewById(R.id.eventDetailsText);

    dateButton.setOnClickListener(new View.OnClickListener() { @Override
        public void onClick(View v) {
            showDatePickerDialog();
        }
    });

    timeButton.setOnClickListener(new View.OnClickListener() { @Override
        public void onClick(View v) {
            showTimePickerDialog();
        }
    });

    saveButton.setOnClickListener(new View.OnClickListener() { @Override
        public void onClick(View v) {
            if (selectedYear != 0 && selectedHour != 0) {
                String eventDetails = "Event Date: " + selectedDay + "/" + (selectedMonth + 1) + "/"
+ selectedYear +
                    "\nEvent Time: " + String.format("%02d:%02d", selectedHour,
selectedMinute);
                eventDetailsText.setText(eventDetails);
            } else {
                Toast.makeText(MainActivity.this, "Please select both date and time!",
Toast.LENGTH_SHORT).show();
            }
        }
    });
}

private void showDatePickerDialog() {
    final Calendar calendar = Calendar.getInstance(); int year =
    calendar.get(Calendar.YEAR); int month = calendar.get(Calendar.MONTH);
    int day = calendar.get(Calendar.DAY_OF_MONTH);

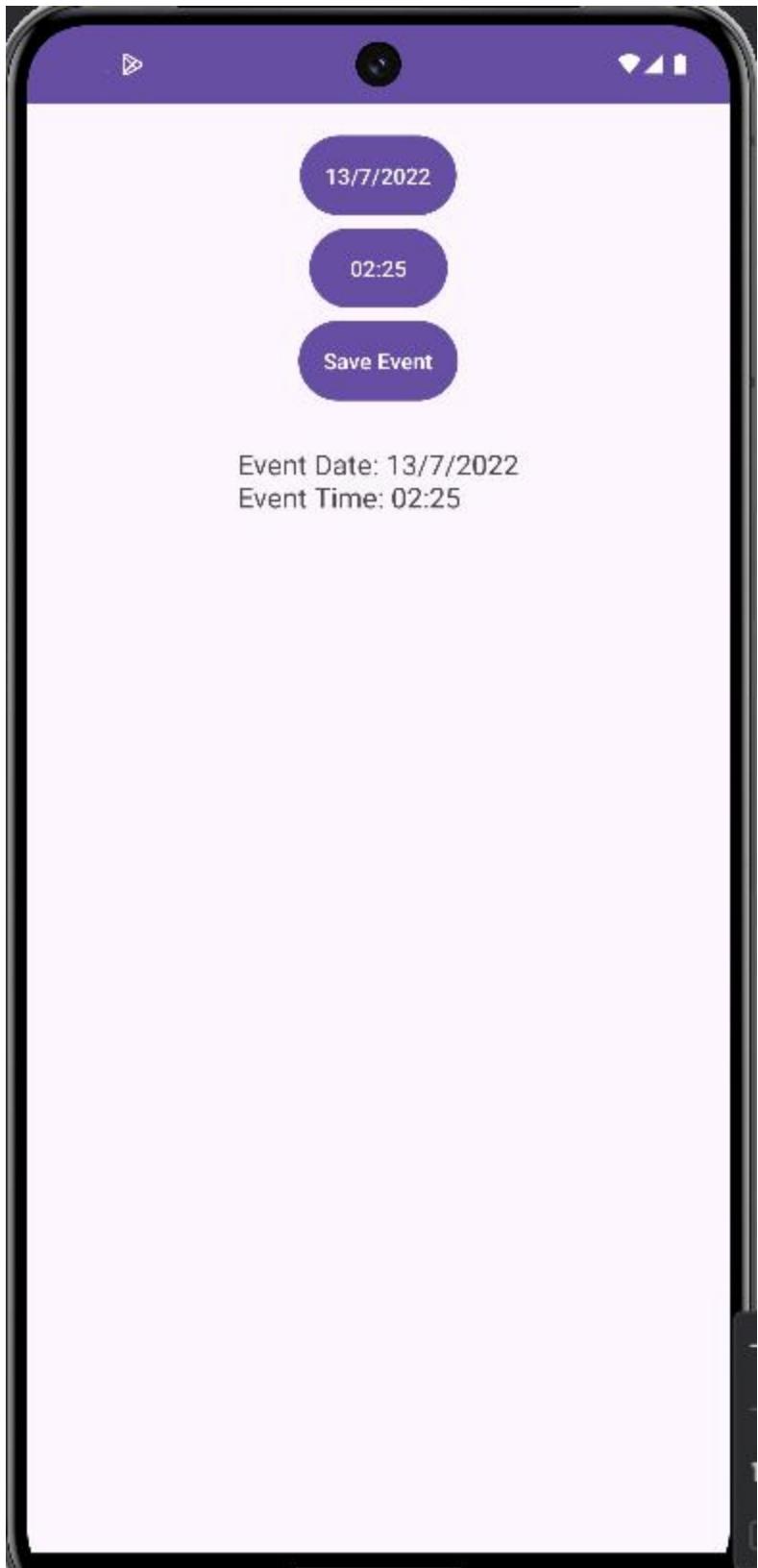
    DatePickerDialog datePickerDialog = new DatePickerDialog(this, (view, year1, month1,
dayOfMonth) -> {
        selectedYear = year1;
        selectedMonth = month1;
        selectedDay = dayOfMonth;
    });
}
```

```
        dateButton.setText(selectedDay + "/" + (selectedMonth + 1) + "/" + selectedYear);
    }, year, month, day);
datePickerDialog.show();
}

private void showTimePickerDialog() {
    final Calendar calendar = Calendar.getInstance(); int hour =
    calendar.get(Calendar.HOUR_OF_DAY); int minute =
    calendar.get(Calendar.MINUTE);

    TimePickerDialog timePickerDialog = new TimePickerDialog(this, (view, hourOfDay, minute1)
        ->{
            selectedHour = hourOfDay;
            selectedMinute = minute1;
            timeButton.setText(String.format("%02d:%02d", selectedHour,
selectedMinute));
        }, hour, minute, true);
    timePickerDialog.show();
}
}
```

Output:



5. Create a media player application that allows the user to play both audio and video files.

The app should display a static image using an ImageView and load an image from a URL using the Glide or Picasso library. Implement buttons for playing an audio file and playing a video file from a URL or local storage. Provide basic controls like play and pause.

Solution:

Xml code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:scaleType="centerCrop"
        android:src="@drawable/staticimage" />

    <Button
        android:id="@+id/loadImageButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Load Image from URL" />

    <Button
        android:id="@+id/playAudioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Play Audio" />

    <Button
        android:id="@+id/pauseAudioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pause Audio" />

    <VideoView
        android:id="@+id/videoView"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:layout_marginTop="16dp" />

    <Button
        android:id="@+id/playVideoButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Play Video" />

    <Button
```

```
    android:id="@+id/pauseVideoButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Pause Video" />
</LinearLayout>
```

Java code:

```
package com.example.la3q5;

import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;
import android.widget.VideoView;
import androidx.appcompat.app.AppCompatActivity; import
com.bumptech.glide.Glide; import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    private ImageView imageView;
    private Button loadImageButton, playAudioButton, pauseAudioButton, playVideoButton,
    pauseVideoButton;
    private MediaPlayer mediaPlayer;
    private VideoView videoView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        imageView = findViewById(R.id.imageView); loadImageButton =
        findViewById(R.id.loadImageButton); playAudioButton =
        findViewById(R.id.playAudioButton); pauseAudioButton =
        findViewById(R.id.pauseAudioButton); playVideoButton =
        findViewById(R.id.playVideoButton); pauseVideoButton =
        findViewById(R.id.pauseVideoButton); videoView = findViewById(R.id.videoView);

        loadImageButton.setOnClickListener(v -> loadImageFromUrl());

        playAudioButton.setOnClickListener(v -> playAudio());
        pauseAudioButton.setOnClickListener(v -> pauseAudio());

        playVideoButton.setOnClickListener(v -> playVideo());
        pauseVideoButton.setOnClickListener(v -> pauseVideo());
    }

    private void loadImageFromUrl() {
```

```

        String imageUrl =
"https://drive.google.com/uc?export=download&id=1BJzZaJW8SGz_kd_0Z48A5bZo_2HAPp
K1\n";
        Glide.with(this)
            .load(imageUrl)
            .placeholder(R.drawable.staticimage)
            .error(R.drawable.error)
            .into(imageView);
    }

private void playAudio() {
    if (mediaPlayer == null) {
        mediaPlayer = new MediaPlayer();
        try {
mediaPlayer.setDataSource("https://drive.google.com/uc?export=download&id=1l2Qj
yGJrunx_32vu243JdtO5tQxiZvMy\n");
            mediaPlayer.prepare();
        } catch (IOException e) { e.printStackTrace(); Toast.makeText(this,
            "Error playing audio",
Toast.LENGTH_SHORT).show();
        }
    }
    mediaPlayer.start();
}

private void pauseAudio() {
    if (mediaPlayer != null && mediaPlayer.isPlaying()) { mediaPlayer.pause();
}

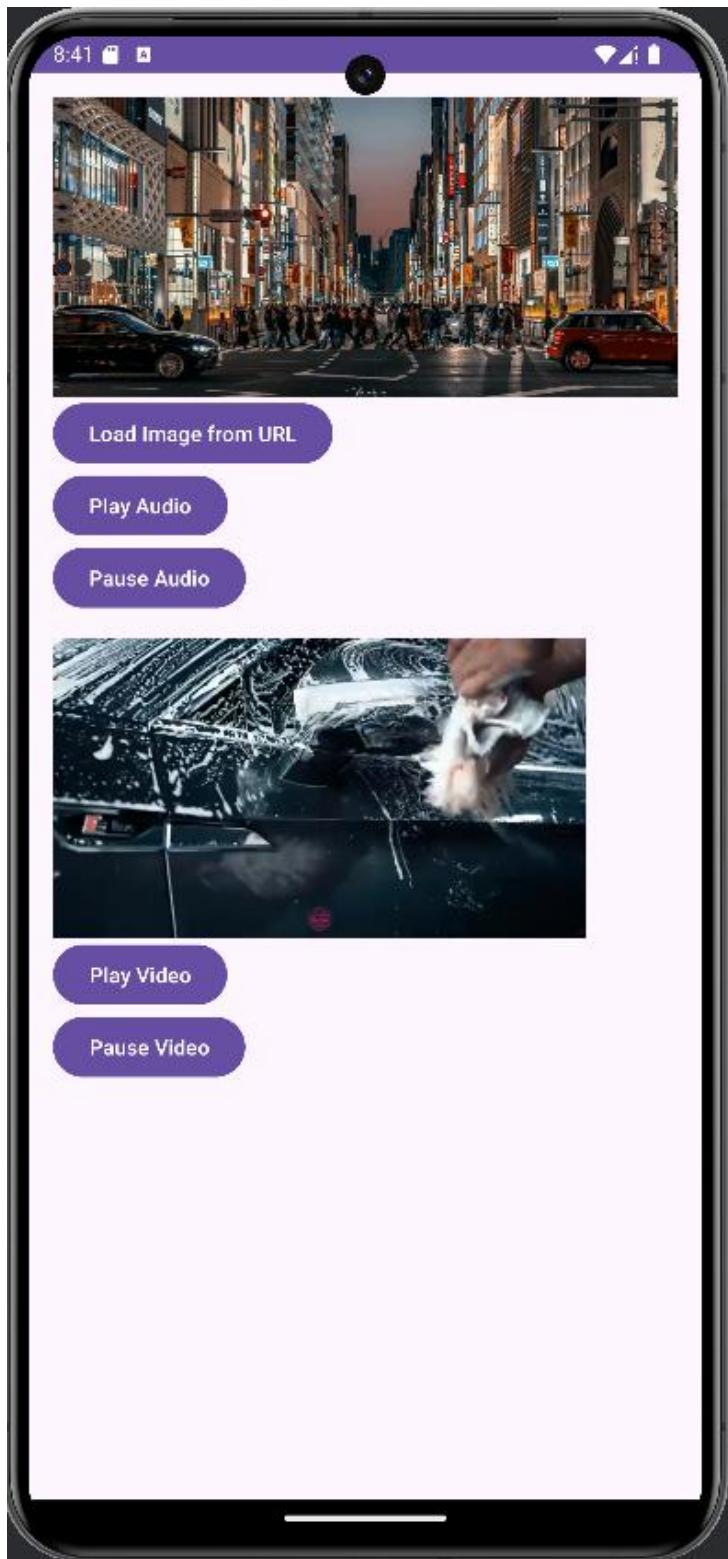
private void playVideo() {
    String videoUrl =
"https://drive.google.com/uc?export=download&id=1bsMaCpogH1rr9XTQNZZoIA_oOHG77c uU\n";
    videoView.setVideoURI(Uri.parse(videoUrl)); videoView.start();
}

private void pauseVideo() {
    if (videoView.isPlaying()) {
        videoView.pause();
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    if (mediaPlayer != null) {
        mediaPlayer.release();
        mediaPlayer = null;
    }
}
}

```

Output:



6. Develop a photo gallery application where users can scroll through a collection of images. Use a ScrollView to display the images in a vertical scroll layout. Additionally, implement a ViewFlipper that cycles through the images automatically or on button click. Lastly, create a tabbed interface with two tabs: "Photos" and "Videos". Each tab should display relevant content.

Solution:

Xml code:

activity_main:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

        <com.google.android.material.tabs.TabLayout
            android:id="@+id/tabLayout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:tabIndicatorColor="@color/design_default_color_primary" />
    </com.google.android.material.appbar.AppBarLayout>

    <androidx.viewpager2.widget.ViewPager2
        android:id="@+id/viewPager"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior" />
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

fragment_photos:

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">

        <ViewFlipper
            android:id="@+id/viewFlipper"
            android:layout_width="match_parent"
```

```

        android:layout_height="200dp"
        android:layout_gravity="center"
        android:autoStart="true"
        android:flipInterval="2000">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/gfglogo" />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/hackerranklogo" />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/leetcodeLogo" />
</ViewFlipper>

<Button
    android:id="@+id/nextButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Next Image" />

<ImageView
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_marginTop="16dp"
    android:src="@drawable/australia" />

<ImageView
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_marginTop="16dp"
    android:src="@drawable/brazil" />

<ImageView
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_marginTop="16dp"
    android:src="@drawable/argentina" />

<ImageView
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_marginTop="16dp"
    android:src="@drawable/india" />

</LinearLayout>
</ScrollView>

```

Fragment_videos:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```

        android:orientation="vertical"
        android:padding="16dp">

    <VideoView
        android:id="@+id/videoView"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:layout_marginBottom="16dp" />

    <Button
        android:id="@+id/playVideoButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Play Video" />
</LinearLayout>

```

Java code:

MainActivity:

```

package com.example.la3q6;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.viewpager2.widget.ViewPager2;
import com.google.android.material.tabs.TabLayout;
import com.google.android.material.tabs.TabLayoutMediator;

public class MainActivity extends AppCompatActivity {

    private TabLayout tabLayout;
    private ViewPager2 viewPager;
    private TabAdapter tabAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tabLayout = findViewById(R.id.tabLayout);    viewPager =
        findViewById(R.id.viewPager);

        tabAdapter = new TabAdapter(this);
        viewPager.setAdapter(tabAdapter);

        new TabLayoutMediator(tabLayout, viewPager, (tab, position) -> { if (position == 0) {
            tab.setText("Photos");
        } else {
            tab.setText("Videos");
        }
    }).attach();
}
}

```

TabAdapter:

```
package com.example.la3q6;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentActivity;
import androidx.viewpager2.adapter.FragmentStateAdapter;

public class TabAdapter extends FragmentStateAdapter {

    public TabAdapter(@NonNull FragmentActivity fragmentActivity) {
        super(fragmentActivity);
    }

    @NonNull
    @Override
    public Fragment createFragment(int position) { switch (position) {
        case 0:
            return new PhotosFragment();
        case 1:
            return new VideosFragment();
        default:
            return new PhotosFragment();
    }
}

    @Override
    public int getItemCount() {
        return 2;
    }
}
```

PhotosFragment:

```
package com.example.la3q6;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ViewFlipper;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class PhotosFragment extends Fragment {

    private ViewFlipper viewFlipper;
    private Button nextButton;

    @Nullable
```

@Override

```

    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
    @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_photos, container, false);

        viewFlipper = view.findViewById(R.id.viewFlipper); nextButton =
        view.findViewById(R.id.nextButton);

        nextButton.setOnClickListener(v -> viewFlipper.showNext());

        return view;
    }
}

```

VedioFragment:

```

package com.example.la3q6;

import android.net.Uri;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.VideoView;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class VideosFragment extends Fragment {

    private VideoView videoView;
    private Button playVideoButton;

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
    @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_videos, container, false);

        videoView = view.findViewById(R.id.videoView); playVideoButton =
        view.findViewById(R.id.playVideoButton);

        playVideoButton.setOnClickListener(v -> playVideo());

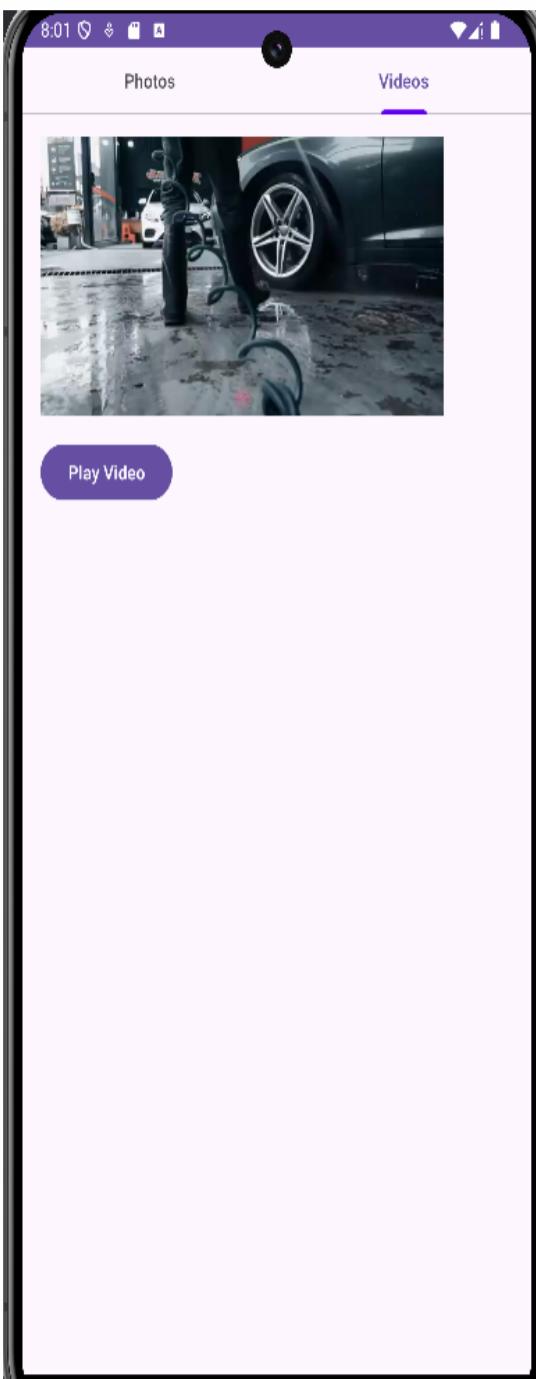
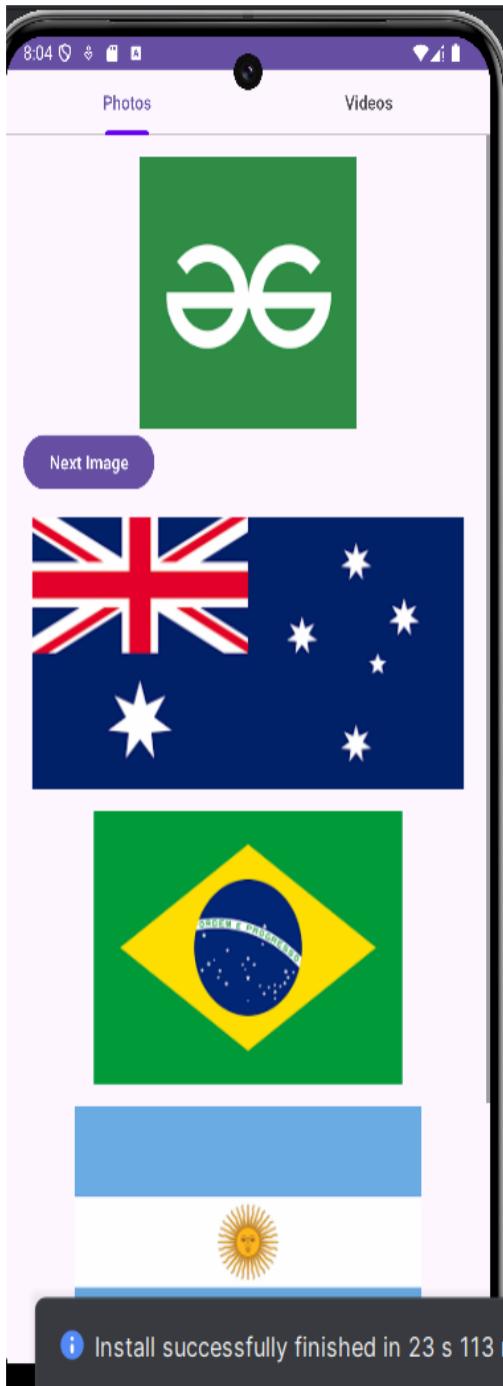
        return view;
    }

    private void playVideo() {
        String videoUrl =
        "https://drive.google.com/uc?export=download&id=1bsMaCpogH1rr9XTQNZZOIA_oOHG77c
        uU\n";
    }
}

```

```
        Uri videoUri = Uri.parse(videoUrl);
        videoView.setVideoURI(videoUri);
        videoView.start();
    }
}
```

Output:



7. Build a notification center application that provides different types of user feedback.

Implement buttons that trigger the following:

A. An AlertDialog asking the user for confirmation (e.g., "Are you sure you want to delete this item?").

B. A Toast that displays a short success message.

C. A PopupWindow that shows additional options when a user clicks on a specific button.

Customize the appearance of each of these UI components.

Solution:

Xml code:

Activity_main:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/alertDialogButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show AlertDialog" />

    <Button
        android:id="@+id/toastButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Toast"
        android:layout_marginTop="20dp" />

    <Button
        android:id="@+id/popupButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show PopupWindow"
        android:layout_marginTop="20dp"/>
</LinearLayout>
```

Popup_layout:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp"
    android:background="@android:color/white">
```



```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Additional Options"
        android:textSize="18sp"
        android:textColor="@android:color/black" />

    <Button
        android:id="@+id/dismissPopupButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Dismiss"
        android:layout_marginTop="10dp"/>
</LinearLayout>

```

Java code:

```

package com.example.la3q7;

import android.os.Bundle;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.PopupWindow;
import android.widget.Toast;
import androidx.appcompat.app.AlertDialog; import
androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private Button alertDialogButton, toastButton, popupButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        alertDialogButton = findViewById(R.id.alertDialogButton); toastButton =
        findViewById(R.id.toastButton); popupButton = findViewById(R.id.popupButton);

        alertDialogButton.setOnClickListener(v -> showAlertDialog());

        toastButton.setOnClickListener(v -> showToast());

        popupButton.setOnClickListener(v -> showPopupWindow(v));
    }

    private void showAlertDialog() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Confirmation");
        builder.setMessage("Are you sure you want to delete this item?");
        builder.setPositiveButton("Yes", (dialog, which) -> {
            Toast.makeText(this, "Item Deleted", Toast.LENGTH_SHORT).show();
        });
        builder.setNegativeButton("No", (dialog, which) -> dialog.dismiss());
    }
}

```

```
        AlertDialog dialog = builder.create();
        dialog.show();
    }

    private void showToast() {
        Toast toast = Toast.makeText(this, "Action completed successfully", Toast.LENGTH_SHORT);
        toast.setGravity(Gravity.CENTER, 0, 0);
        toast.show();
    }

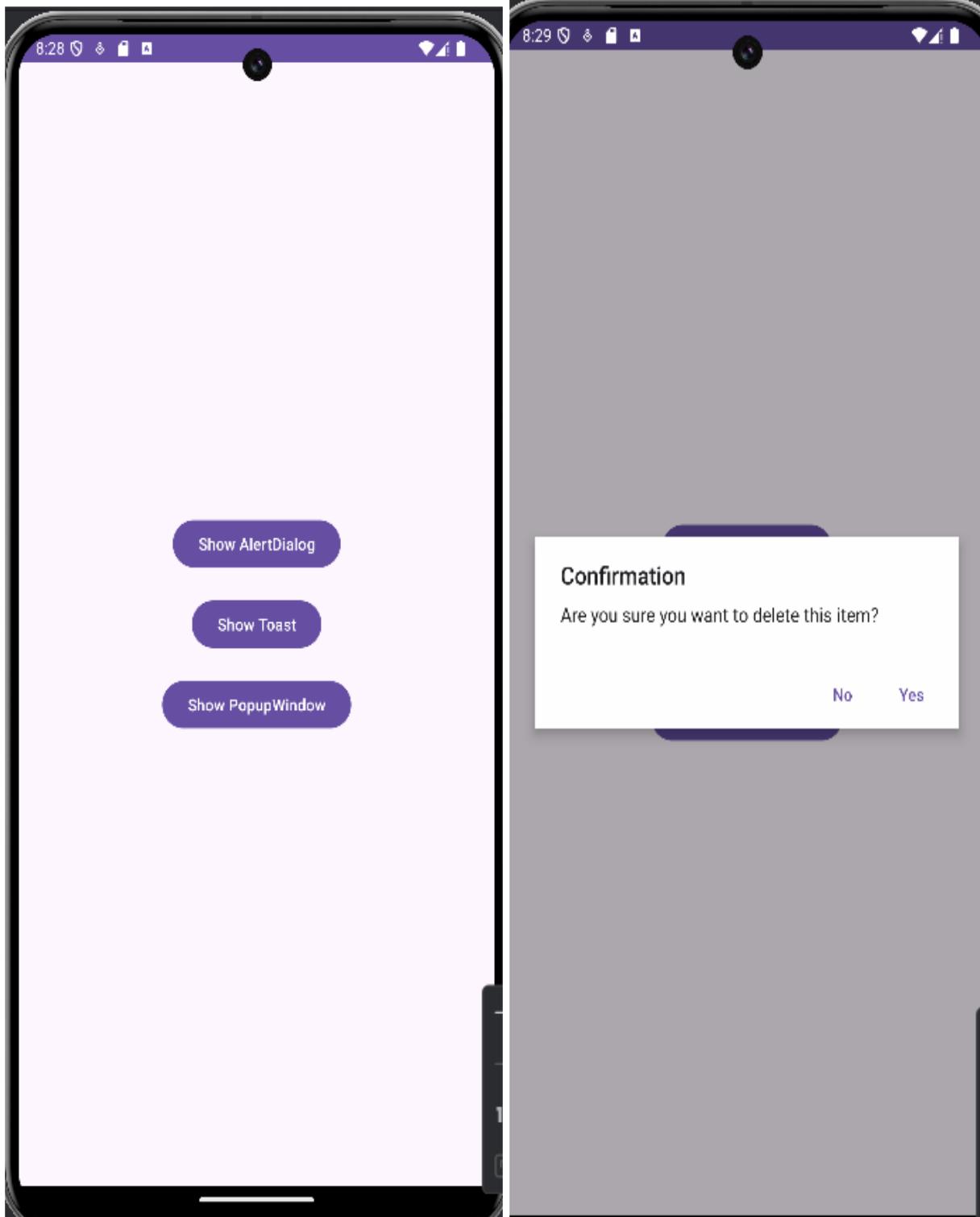
    private void showPopupWindow(View anchorView) { LayoutInflator
        inflater = (LayoutInflater)
getSystemService(LAYOUT_INFLATER_SERVICE);
        View popupView = inflater.inflate(R.layout.popup_layout, null);

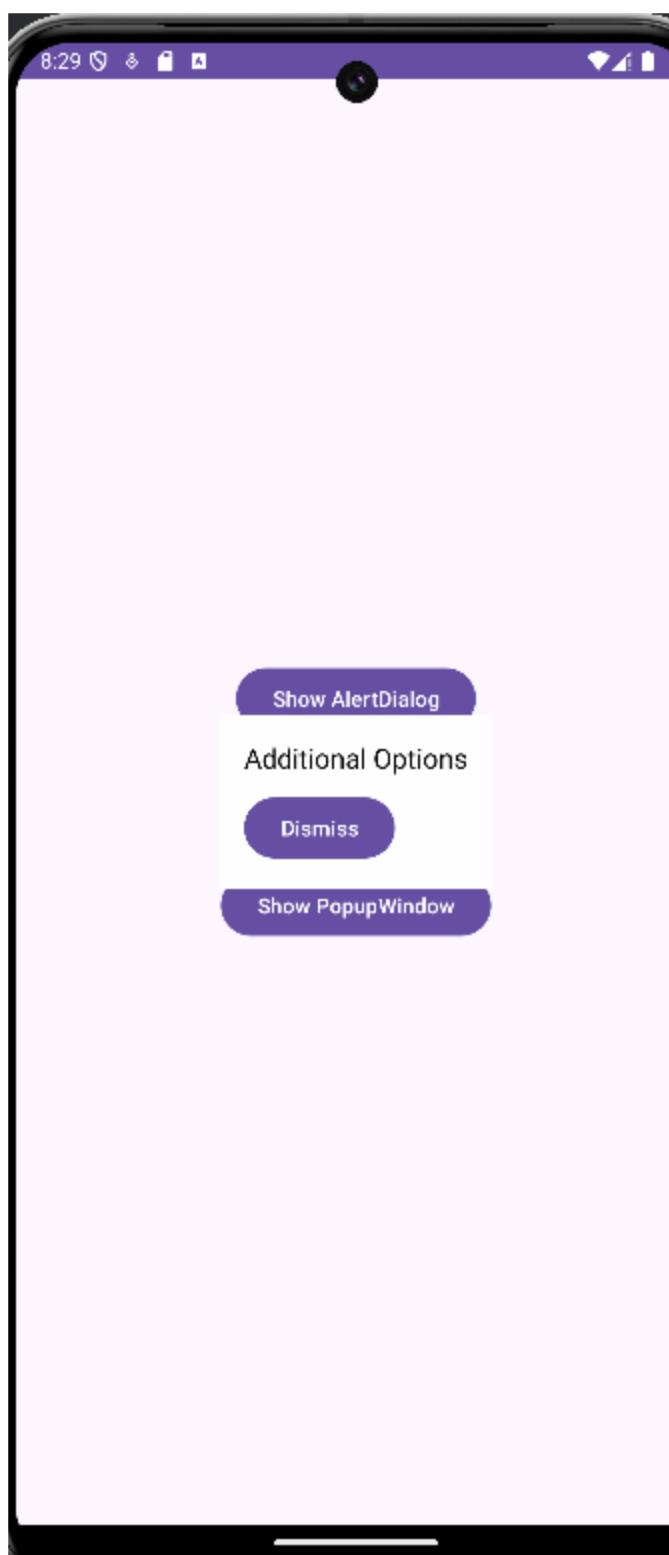
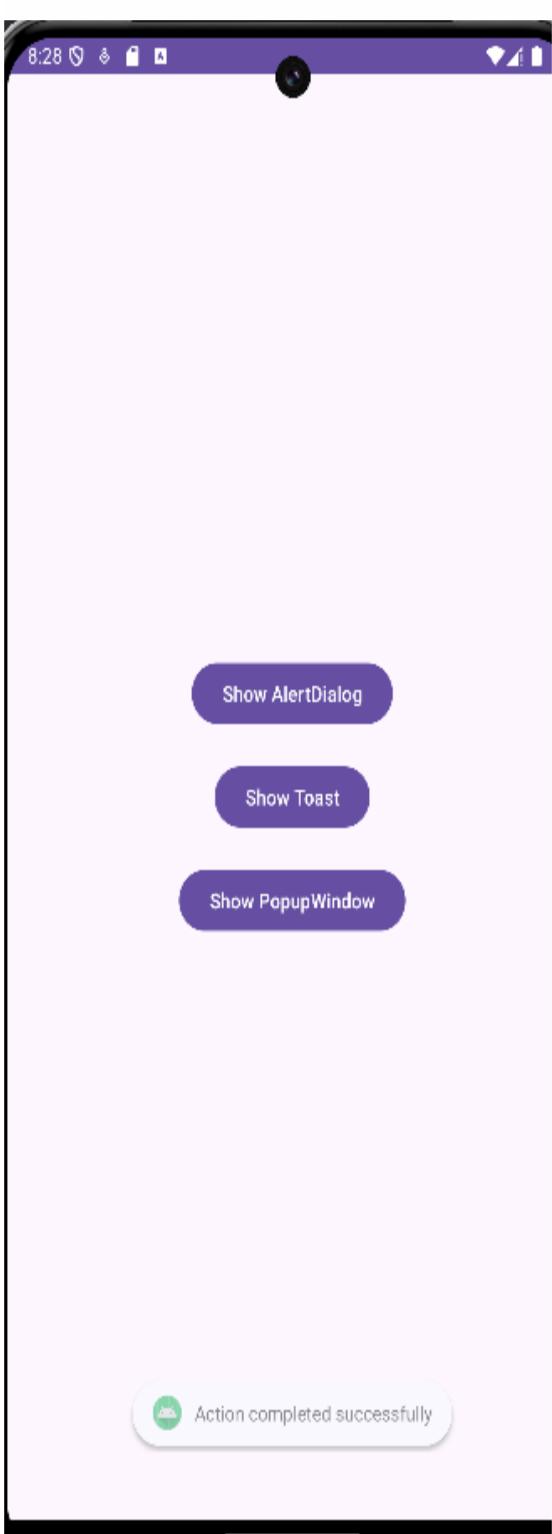
        final PopupWindow popupWindow = new PopupWindow(
            popupView,
            ViewGroup.LayoutParams.WRAP_CONTENT,
            ViewGroup.LayoutParams.WRAP_CONTENT,
            true
        );

        popupWindow.showAtLocation(anchorView, Gravity.CENTER, 0, 0);

        Button dismissButton = popupView.findViewById(R.id.dismissPopupButton);
        dismissButton.setOnClickListener(v -> popupWindow.dismiss());
    }
}
```

Output:





8. Design a mobile application with a navigation bar at the top. The app should have an options menu that includes items like "Settings", "Profile", and "Logout". Additionally, implement a context menu that appears when a user long presses on an item in a Recycler View. Create a button that displays a popup menu with options like "Edit" "Share" and "Delete" when clicked.

Solution:

Xml code:

Activity_main:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr actionBarSize"
        android:background="?attr colorPrimary"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
        app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginTop="8dp"
        app:layout_constraintTop_toBottomOf="@+id/toolbar"
        app:layout_constraintBottom_toTopOf="@+id/popup_button" />

    <Button
        android:id="@+id/popup_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Popup Menu"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/recyclerView" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Context_menu:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
```

```
<item android:id="@+id/context_edit"
      android:title="Edit"/>
<item android:id="@+id/context_share"
      android:title="Share"/>
<item android:id="@+id/context_delete"
      android:title="Delete"/>
</menu>
```

Options_menu:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"> <item
    android:id="@+id/settings"
    android:title="Settings"/>
<item android:id="@+id/profile"
    android:title="Profile"/>
<item android:id="@+id/logout"
    android:title="Logout"/>
</menu>
```

Popup_menu:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"> <item
    android:id="@+id/edit"
    android:title="Edit"/>
<item android:id="@+id/share"
    android:title="Share"/>
<item android:id="@+id/delete"
    android:title="Delete"/>
</menu>
```

Java code:

MainActivity:

```
package com.example.la3q8;

import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity; import
androidx.appcompat.widget.PopupMenu; import
androidx.appcompat.widget.Toolbar;
import androidx.recyclerview.widget.LinearLayoutManager; import
androidx.recyclerview.widget.RecyclerView;
```

```
public class MainActivity extends AppCompatActivity {
```

```
private int selectedPosition;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Toolbar toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    RecyclerView recyclerView = findViewById(R.id.recyclerView);
    recyclerView.setLayoutManager(new LinearLayoutManager(this)); MyAdapter adapter =
    new MyAdapter(); recyclerView.setAdapter(adapter);

    registerForContextMenu(recyclerView);

    Button popupButton = findViewById(R.id.popup_button);
    popupButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            showPopupMenu(v);
        }
    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) { MenuInflater
    inflater = getMenuInflater();
    inflater.inflate(R.menu.options_menu, menu); return true;
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) { int id =
    item.getItemId();

    if (id == R.id.settings) {
        Toast.makeText(this, "Settings clicked", Toast.LENGTH_SHORT).show(); return true;
    } else if (id == R.id.profile) {
        Toast.makeText(this, "Profile clicked", Toast.LENGTH_SHORT).show(); return true;
    } else if (id == R.id.logout) {
        Toast.makeText(this, "Logout clicked", Toast.LENGTH_SHORT).show(); return true;
    }

    return super.onOptionsItemSelected(item);
}

private void showPopupMenu(View view) { PopupMenu popup = new
    PopupMenu(this, view);
    popup.getMenuInflater().inflate(R.menu.popup_menu, popup.getMenu());

    popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener()
{
```

```
    @Override
    public boolean onMenuItemClick(MenuItem item) { int id =
        item.getItemId();

        if (id == R.id.edit) {
            Toast.makeText(MainActivity.this, "Edit clicked",
Toast.LENGTH_SHORT).show();
            return true;
        } else if (id == R.id.share) {
            Toast.makeText(MainActivity.this, "Share clicked",
Toast.LENGTH_SHORT).show();
            return true;
        } else if (id == R.id.delete) {
            Toast.makeText(MainActivity.this, "Delete clicked",
Toast.LENGTH_SHORT).show();
            return true;
        }

        return false;
    }
});

popup.show();
}

@Override
public void onCreateContextMenu(@NonNull android.view.ContextMenu menu, @NonNull View v, @NonNull android.view.ContextMenu.ContextMenuItemInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    getMenuInflater().inflate(R.menu.context_menu, menu);
}

@Override
public boolean onContextItemSelected(@NonNull MenuItem item) { int id =
    item.getItemId();

    if (id == R.id.context_edit) {
        Toast.makeText(this, "Context Edit clicked",
Toast.LENGTH_SHORT).show();
        return true;
    } else if (id == R.id.context_share) {
        Toast.makeText(this, "Context Share clicked",
Toast.LENGTH_SHORT).show();
        return true;
    } else if (id == R.id.context_delete) {
        Toast.makeText(this, "Context Delete clicked",
Toast.LENGTH_SHORT).show();
        return true;
    }

    return super.onContextItemSelected(item);
}

public void setSelectedPosition(int position){ selectedPosition =
position;
```


MyAdapter:

```
package com.example.la3q8;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class MyAdapter extends RecyclerView.Adapter<MyAdapter.MyViewHolder> {

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view =
LayoutInflator.from(parent.getContext()).inflate(android.R.layout.simple_list_i
tem_1, parent, false);
        return new MyViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {
        holder.textView.setText("Item " + position);
    }

    @Override
    public int getItemCount() {
        return 10;
    }

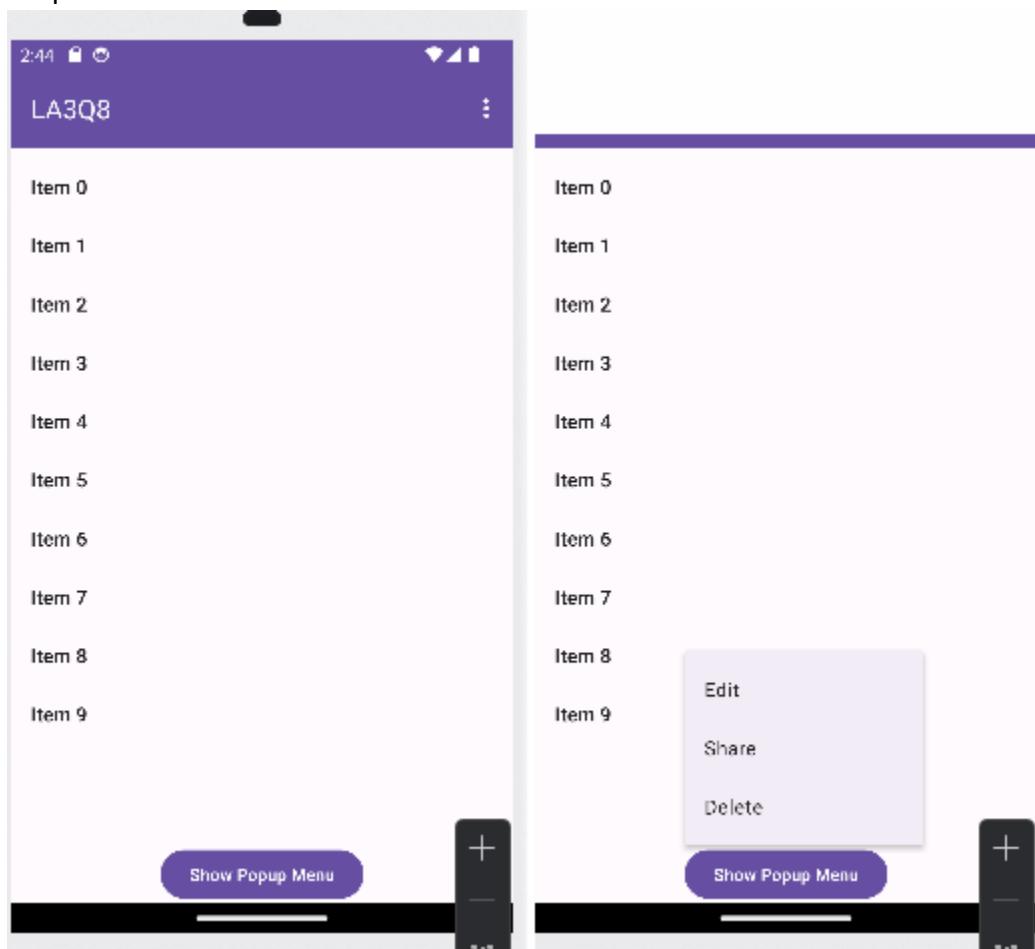
    public static class MyViewHolder extends RecyclerView.ViewHolder implements View.OnLongClickListener {

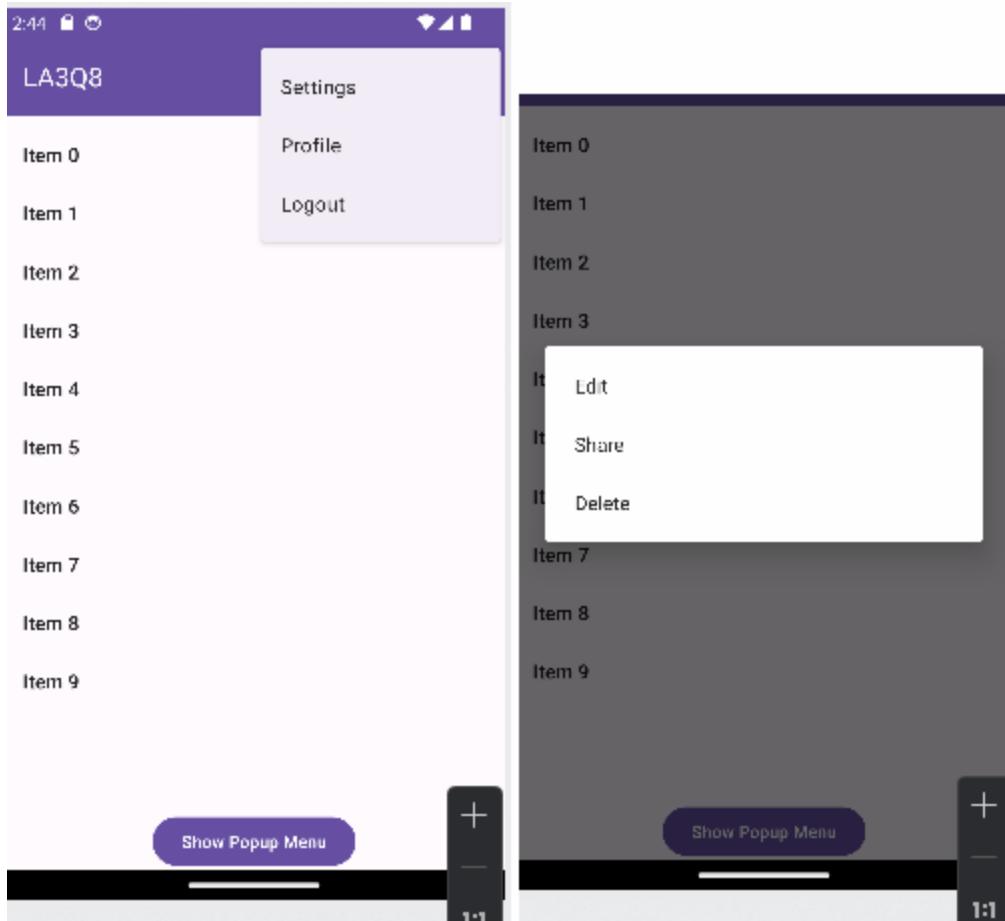
        TextView textView;

        public MyViewHolder(@NonNull View itemView) {
            super(itemView);
            textView = itemView.findViewById(android.R.id.text1);
            itemView.setOnLongClickListener(this);
        }

        @Override
        public boolean onLongClick(View v) {
            ((MainActivity)
v.getContext()).setSelectedPosition(getAdapterPosition()); v.showContextMenu();
            return true;
        }
    }
}
```

Output:





9. Create a feedback form for an e-commerce application. The form should collect the user's name (EditText), email address (EditText), rating for the service (RatingBar), and any additional comments (EditText). Validate that the name and email fields are not left empty and that the email follows a proper format. On form submission, display a confirmation message using a Toast, and reset the form fields.

Solution:

Xml code:

Activity_main:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/name_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your name" />
```

```
        android:inputType="textPersonName" />

<EditText
    android:id="@+id/email_input"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter your email"
    android:inputType="textEmailAddress" />

<RatingBar
    android:id="@+id/rating_bar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:numStars="5"
    android:stepSize="1.0" />

<EditText
    android:id="@+id/comments_input"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Additional comments"
    android:inputType="textMultiLine"
    android:minLines="3" />

<Button
    android:id="@+id/feedback_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Give Feedback" />

</LinearLayout>
```

Activity_feedback:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/name_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your name"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/email_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your email"
        android:inputType="textEmailAddress" />

    <RatingBar
```

```
    android:id="@+id/rating_bar"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:numStars="5"
        android:stepSize="1.0" />

<EditText
    android:id="@+id/comments_input"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Additional comments"
    android:inputType="textMultiLine"
    android:minLines="3" />

<Button
    android:id="@+id/submit_button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Submit Feedback" />

</LinearLayout>

```

Java code:

ActivityMain:

```

package com.example.la3q9;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button feedbackButton = findViewById(R.id.feedback_button);
        feedbackButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,
FeedbackActivity.class);
                startActivity(intent);
            }
        });
    }
}

```

FeedbackActivity:

```
package com.example.la3q9;
```

```
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RatingBar;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class FeedbackActivity extends AppCompatActivity {

    private EditText nameInput, emailInput, commentsInput; private RatingBar
    ratingBar; private Button submitButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_feedback);

        nameInput = findViewById(R.id.name_input); emailInput =
        findViewById(R.id.email_input); ratingBar = findViewById(R.id.rating_bar);
        commentsInput = findViewById(R.id.comments_input); submitButton =
        findViewById(R.id.submit_button);

        submitButton.setOnClickListener(new View.OnClickListener() { @Override
            public void onClick(View v) {
                submitFeedback();
            }
        });
    }

    private void submitFeedback() {
        String name = nameInput.getText().toString().trim(); String email =
        emailInput.getText().toString().trim(); String comments =
        commentsInput.getText().toString().trim(); float rating = ratingBar.getRating();

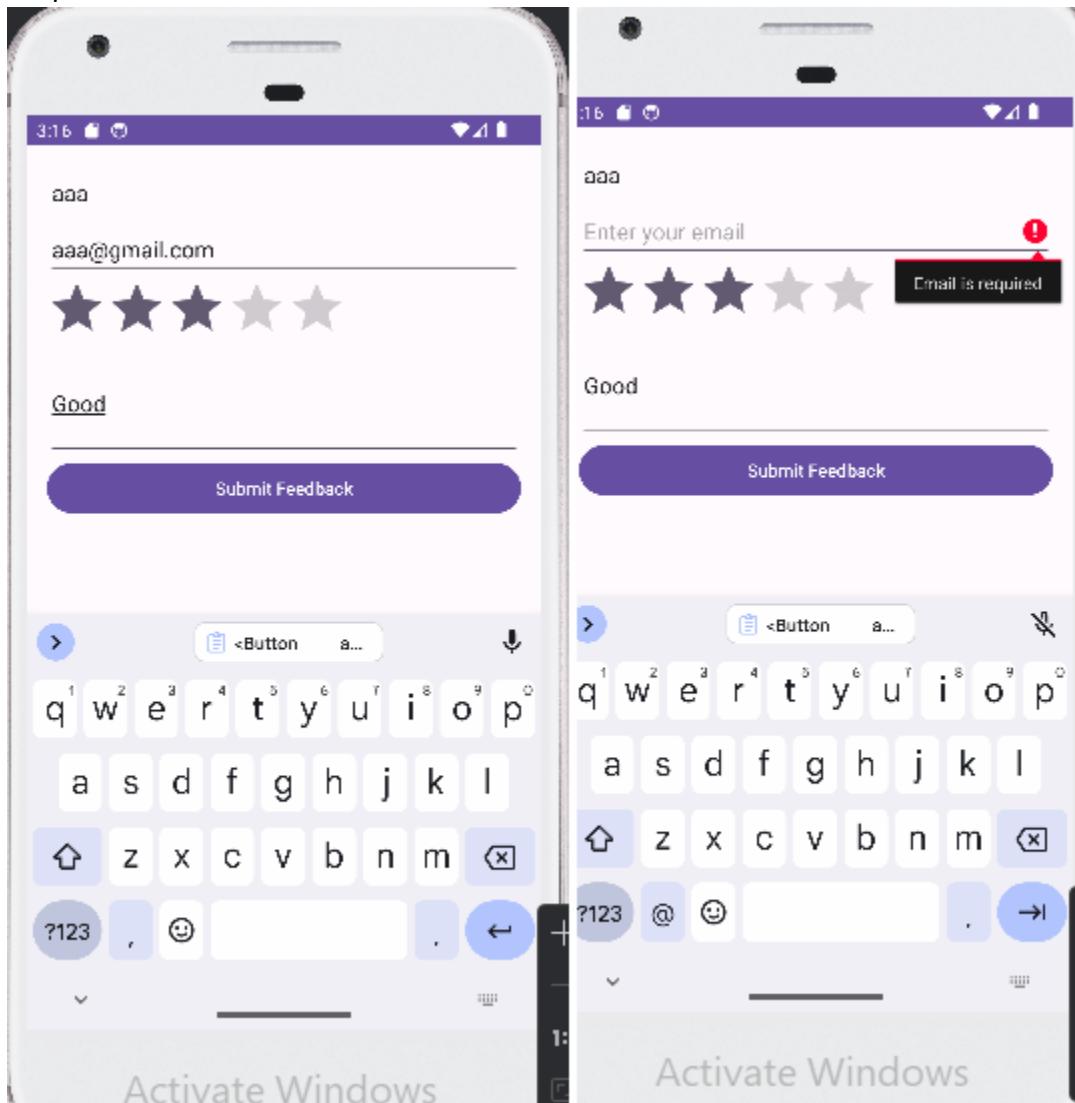
        if (TextUtils.isEmpty(name)) {
            nameInput.setError("Name is required");
            return;
        }

        if (TextUtils.isEmpty(email)) {
            emailInput.setError("Email is required");
            return;
        }

        if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            emailInput.setError("Please enter a valid email"); return;
        }
    }
}
```

```
        Toast.makeText(this, "Thank you for your feedback!",  
Toast.LENGTH_SHORT).show();  
  
        nameInput.setText("");  
        emailInput.setText("");  
        ratingBar.setRating(0);  
        commentsInput.setText("");  
    }  
}
```

Output:



10. Design a reservation form for booking a table at a restaurant. The form should have fields for selecting the number of guests (NumberPicker), date (DatePicker), time (TimePicker), and any special requests (EditText). Once all details are filled out, the user should click a "Reserve" button that displays the entered details in a confirmation AlertDialog. The form should reset after the reservation is confirmed.

Solution:

Xml code:

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="center"
        android:padding="16dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Number of Guests"
            android:textSize="18sp"
            android:layout_marginBottom="8dp" />

        <NumberPicker
            android:id="@+id/numberPickerGuests"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Select Date"
            android:textSize="18sp"
            android:layout_marginTop="16dp"
            android:layout_marginBottom="8dp" />

        <DatePicker
            android:id="@+id/datePicker"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:calendarViewShown="true" />

        <TextView
            android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:text="Select Time"
```

```

        android:textSize="18sp"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="8dp" />

    <TimePicker
        android:id="@+id/timePicker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:timePickerMode="spinner" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Special Requests"
        android:textSize="18sp"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="8dp" />

    <EditText
        android:id="@+id/editTextSpecialRequests"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Any special requests"
        android:minLines="3"
        android:inputType="textMultiLine" />

    <Button
        android:id="@+id/reserve_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Reserve"
        android:layout_marginTop="24dp" />

</LinearLayout>
</ScrollView>

```

Java code:

```

package com.example.la3q10;

import android.os.Bundle;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.NumberPicker;
import android.widget.TimePicker;

import androidx.appcompat.app.AlertDialog; import
androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private NumberPicker numberPickerGuests; private
    DatePicker datePicker; private TimePicker timePicker;
    private EditText editTextSpecialRequests; private Button
    buttonReserve;

```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main); initializeUI();
    setupNumberPicker();
    timePicker.setIs24HourView(true);
    buttonReserve.setOnClickListener(v -> confirmReservation());
}

private void initializeUI() {
    numberPickerGuests = findViewById(R.id.numberPickerGuests); datePicker =
    findViewById(R.id.datePicker); timePicker = findViewById(R.id.timePicker);
    editTextSpecialRequests = findViewById(R.id.editTextSpecialRequests); buttonReserve =
    findViewById(R.id.reserve_button);
}

private void setupNumberPicker() {
    numberPickerGuests.setMinValue(1);
    numberPickerGuests.setMaxValue(20);
}

private void confirmReservation() {
    int numberOfGuests = numberPickerGuests.getValue(); int day =
    datePicker.getDayOfMonth(); int month = datePicker.getMonth() + 1;
    int year = datePicker.getYear();
    int hour = timePicker.getHour();
    int minute = timePicker.getMinute();
    String specialRequests =
    editTextSpecialRequests.getText().toString().trim();

    String reservationDetails = "Guests: " + numberOfGuests + "\n" +
        "Date: " + day + "/" + month + "/" + year + "\n" +
        "Time: " + String.format("%02d:%02d", hour, minute) + "\n" +
        "Special Requests: " + (specialRequests.isEmpty() ? "None" :
    specialRequests);

    new AlertDialog.Builder(this)
        .setTitle("Confirm Reservation")
        .setMessage(reservationDetails)
        .setPositiveButton("Confirm", (dialog, which) -> resetForm())
        .setNegativeButton("Cancel", null)
        .show();
}

private void resetForm() {
    numberPickerGuests.setValue(1);
    datePicker.updateDate(datePicker.getYear(), datePicker.getMonth(),
datePicker.getDayOfMonth());
    timePicker.setHour(12);
    timePicker.setMinute(0);
    editTextSpecialRequests.setText("");
}
}
```

Output:

The image displays two side-by-side screenshots of a mobile application interface for booking a reservation.

Left Screenshot:

- Number of Guests:** A numeric input field showing the value "3".
- Select Date:** A date picker showing "2024 Fri, Nov 8".
- Calendar:** A November 2024 calendar with the 8th highlighted.
- Select Time:** A time picker showing "03 24".
- Time Input:** A numeric input field showing "04 : 25".

Right Screenshot:

- Date Header:** "2024 Fri, Nov 8".
- Calendar:** A November 2024 calendar with the 8th highlighted.
- Select Time:** A time picker showing "03 24".
- Time Input:** A numeric input field showing "04 : 25".
- Special Requests:** A text input field containing "Keep it ready".
- Reserve Button:** A purple button labeled "Reserve".

Select Date

2024

Fri, Nov 8



November 2024



S

M

T

W

T

F

S

1 2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Confirm Reservation

Guests: 4

Date: 8/11/2024

Time: 04:25

Special Requests: Keep it ready

[Cancel](#) [Confirm](#)

03

24

04

25

05

26

Special Requests

package com.example.la3q10; imp...

Keep it ready

[Reserve](#)



Pimpri Chinchwad Education Trust's Pimpri Chinchwad College of Engineering

Assignment-4

Roll No:123M1H010

Name of Student:Harshal Bhamare Submission Date: 16/ 10 /2024

1. Create an Android application that issues a simple notification when a button is clicked. The notification should display a title, message, and small icon. Ensure that the notification appears in the status bar and can be dismissed by the user. Use the NotificationCompat.Builder class to build and issue the notification.

Solution:

xml code:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/notifyButton"
        android:layout_width="220dp"
        android:layout_height="89dp"
        android:text="Notification"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.529"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.822" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

java code:

```
package com.example.la4q1;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Context;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    private static final String CHANNEL_ID = "notify_001";
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    createNotificationChannel();
    requestNotificationPermission();

    Button notifyButton = findViewById(R.id.notifyButton);
    notifyButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            showNotification();
        }
    });
}

private void showNotification() {
    NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
        .setSmallIcon(R.drawable.notification) // Ensure to add a small
icon in res/drawable
        .setContentTitle("Notification")
        .setContentText("This is a notification")
        .setPriority(NotificationCompat.PRIORITY_DEFAULT);

    NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
    notificationManager.notify(0, builder.build());
}

private void createNotificationChannel() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        CharSequence name = "Notification Channel";
        String description = "Channel for simple notifications";
        int importance = NotificationManager.IMPORTANCE_DEFAULT;
        NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
name, importance);
        channel.setDescription(description);

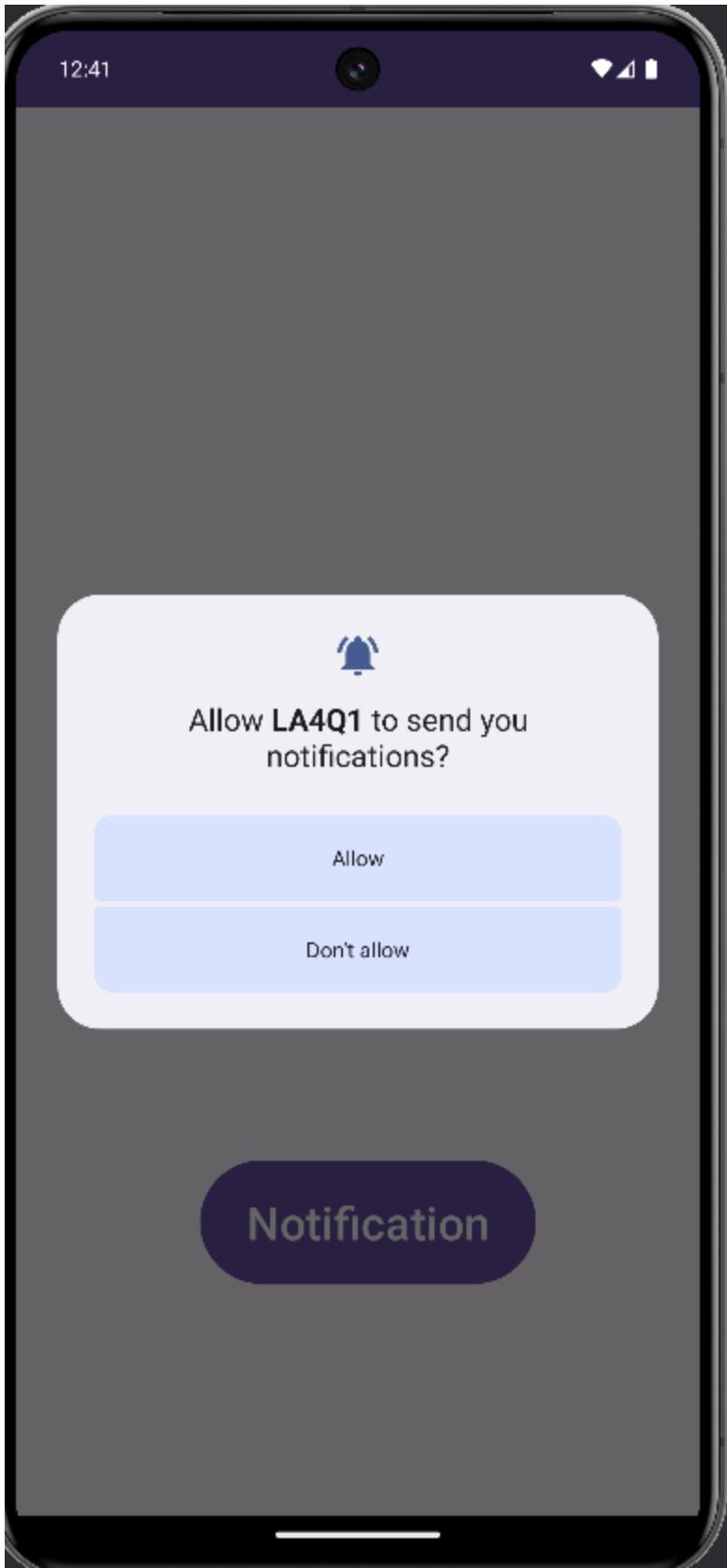
        NotificationManager notificationManager =
getSystemService(NotificationManager.class);
        notificationManager.createNotificationChannel(channel);
    }
}

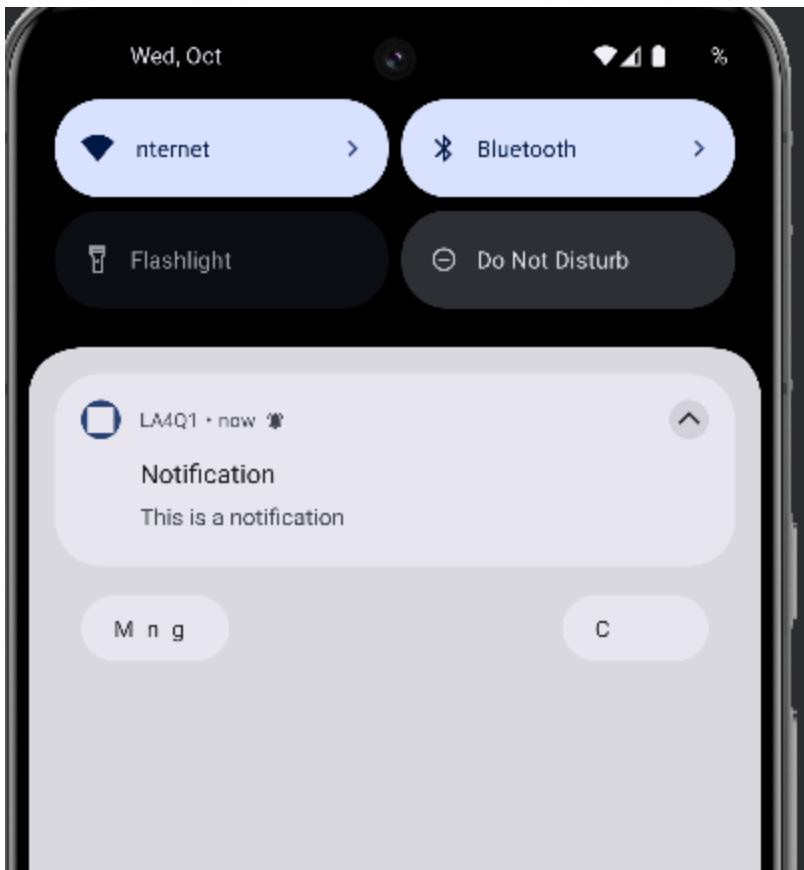
private void requestNotificationPermission() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
        if
(checkSelfPermission(android.Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {
            requestPermissions(new
String[] { android.Manifest.permission.POST_NOTIFICATIONS }, 1);
        }
    }
}

```

Output:







2. Design an app that triggers a basic notification with a clickable action. The notification should have a "View" button that, when clicked, opens a specific activity within the app. Use an Intent to handle the notification action, and display the action's result within the new activity.

Solution:

xml code:

activity_main:

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">

    <Button
        android:id="@+id/btn_notify"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Trigger Notification" />

</LinearLayout>
```

activity_view:

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">
```

```

<Button
    android:id="@+id	btn_notify"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Trigger Notification" />

</LinearLayout>

```

java code:

MainActivity:

```

package com.example.la4q2;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

public class MainActivity extends AppCompatActivity {

    private static final String CHANNEL_ID = "my_channel_id";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button notifyButton = findViewById(R.id.btn_notify);
        notifyButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                triggerNotification();
            }
        });
    }

    private void triggerNotification() {
        Intent intent = new Intent(this, ViewActivity.class);
        intent.putExtra("EXTRA_MESSAGE", "This is the action result from notification.");

        PendingIntent pendingIntent = PendingIntent.getActivity(
            this, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT |
        PendingIntent.FLAG_IMMUTABLE);

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel channel = new NotificationChannel(
                CHANNEL_ID, "My Channel",
            NotificationManager.IMPORTANCE_DEFAULT);
            NotificationManager manager =
            getSystemService(NotificationManager.class);
            manager.createNotificationChannel(channel);
        }
    }
}

```

```

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
        .setSmallIcon(R.drawable.noti)
        .setContentTitle("New Notification")
        .setContentText("Click to view the action result.")
        .setPriority(NotificationCompat.PRIORITY_DEFAULT)
        .setContentIntent(pendingIntent)
        .addAction(R.drawable.hackerranklogo, "View", pendingIntent)
        .setAutoCancel(true);

        NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(1, builder.build());
    }
}

```

ViewActivity:

```

package com.example.la4q2;

import android.os.Bundle;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class ViewActivity extends AppCompatActivity {

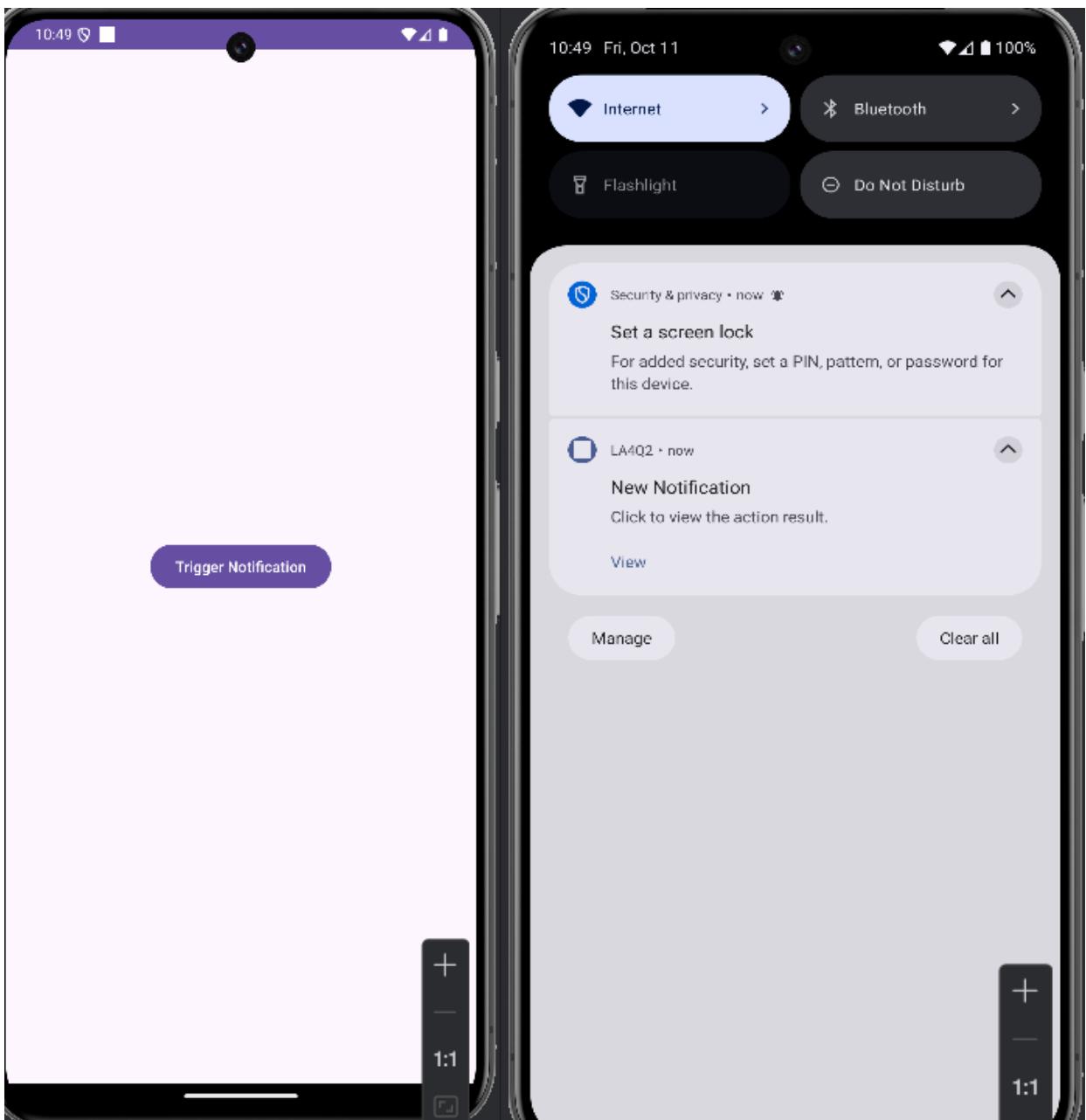
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view);

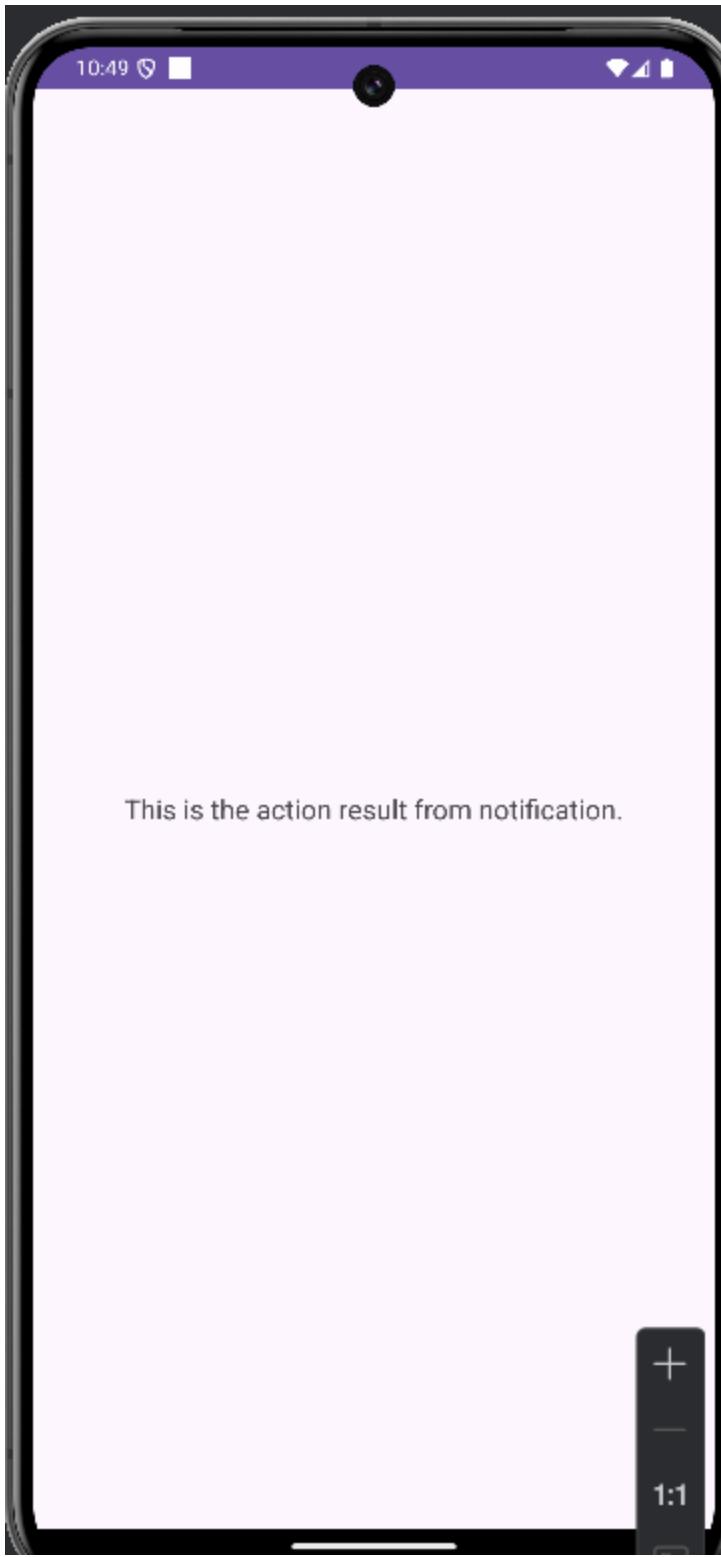
        TextView resultText = findViewById(R.id.tv_result);

        String message = getIntent().getStringExtra("EXTRA_MESSAGE");
        if (message != null) {
            resultText.setText(message);
        }
    }
}

```

Output:





3. Create an Android application that triggers a simple notification when a button is clicked. Use the `NotificationCompat.Builder` class to build the notification and set its properties, such as title, text, and icon. Ensure that the notification appears in the status bar and can be expanded to show additional content.

Solution:

xml code:

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/notifyButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Notify Me"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

java code:

```

package com.example.la4q3;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;

import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    private final String CHANNEL_ID = "channel_id_example";
    private final int NOTIFICATION_ID = 001;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        createNotificationChannel();

        Button notifyButton = findViewById(R.id.notifyButton);
        notifyButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                triggerNotification();
            }
        });
    }

    private void triggerNotification() {
        Intent intent = new Intent(this, MainActivity.class);

```

```

        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent,
PendingIntent.FLAG_IMMUTABLE);

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
        .setSmallIcon(R.drawable.notification)
        .setContentTitle("Simple Notification")
        .setContentText("This is a simple notification")
        ..setStyle(new NotificationCompat.BigTextStyle()
        .bigText("This is an expandable notification. Android
programming is the best"))
        .setPriority(NotificationCompat.PRIORITY_DEFAULT)
        .setContentIntent(pendingIntent)
        .setAutoCancel(true);

        NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(this);
        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.POST_NOTIFICATIONS) !=

PackageManager.PERMISSION_GRANTED) {

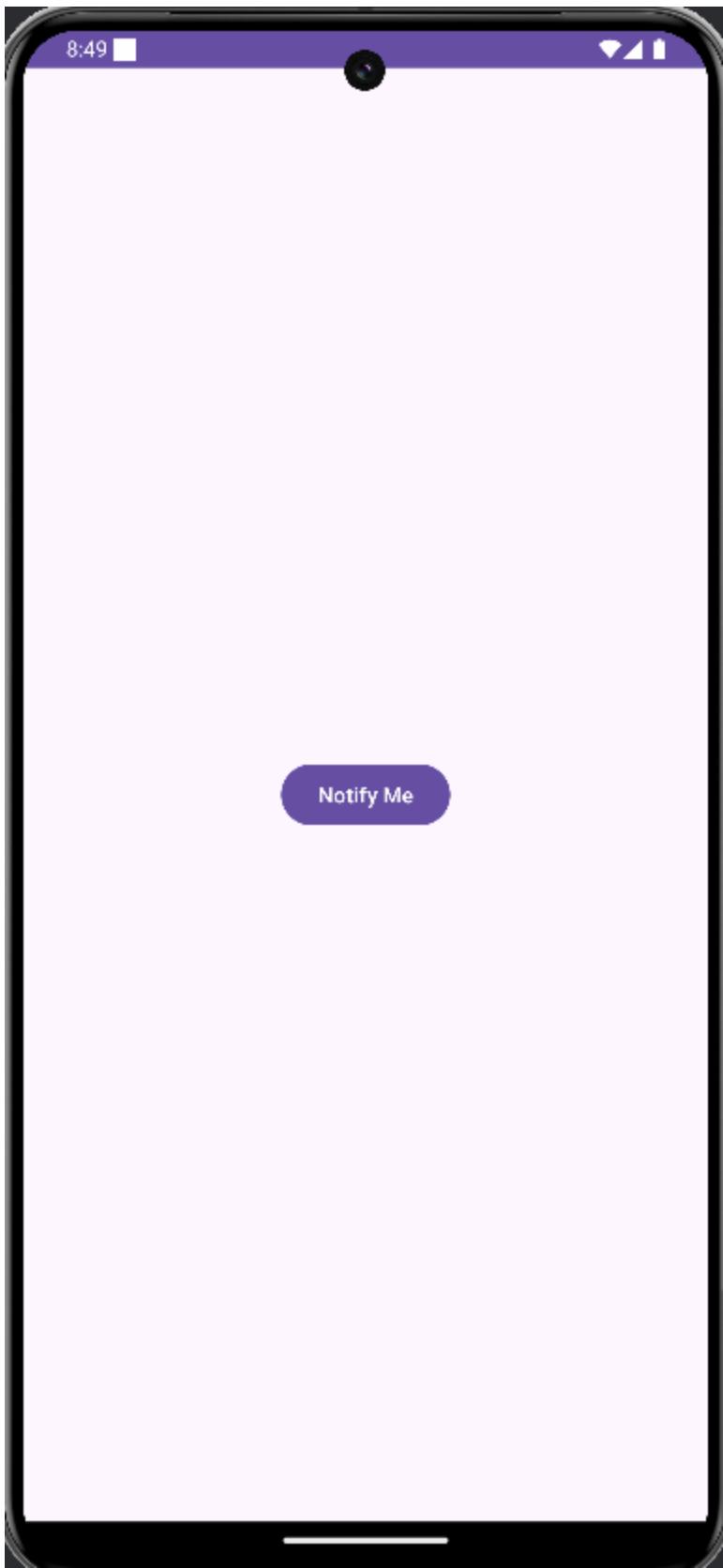
            return;
        }
        notificationManager.notify(NOTIFICATION_ID, builder.build());
    }

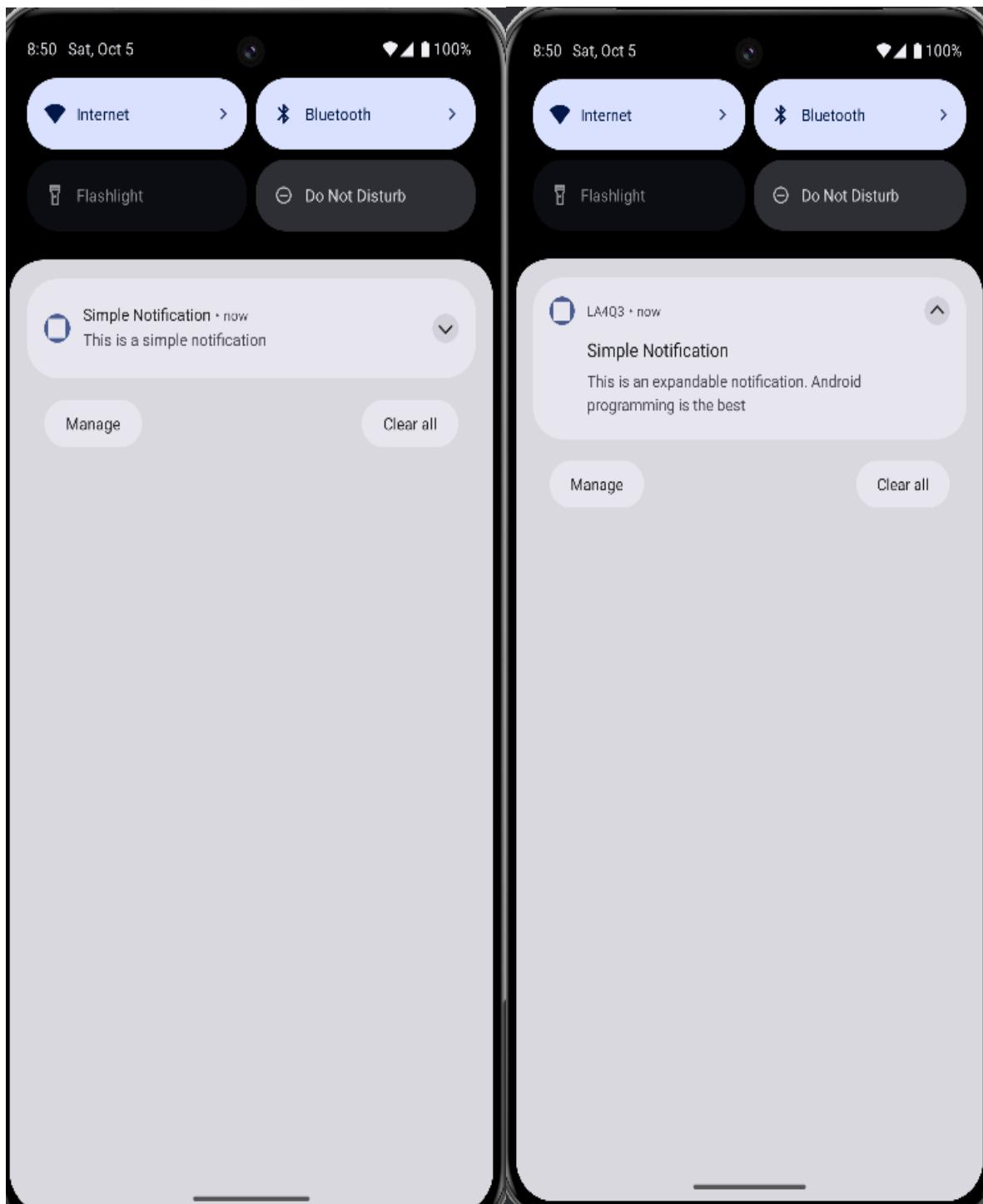
    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            CharSequence name = "Example Channel";
            String description = "This is a channel for example notifications";
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
name, importance);
            channel.setDescription(description);

            NotificationManager notificationManager =
getSystemService(NotificationManager.class);
            notificationManager.createNotificationChannel(channel);
        }
    }
}

```

Output:





4. Build an application that generates a notification with custom properties such as sound, vibration, and LED light color. Use the `NotificationCompat.Builder` class to

set these properties. The app should allow the user to configure these properties through a settings screen and preview the notification with the chosen settings.

Solution:

xml code:

activity_main:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">

    <Button
        android:id="@+id/openSettingsButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Open Settings"
        app:layout_constraintBottom_toTopOf="@+id/previewButton"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.605" />

    <Button
        android:id="@+id/previewButton"
        android:layout_width="199dp"
        android:layout_height="77dp"
        android:layout_marginBottom="204dp"
        android:text="Preview Notification"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_settings:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".SettingsActivity">

    <androidx.appcompat.widget.SwitchCompat
        android:id="@+id/vibrationSwitch"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:minHeight="32dp"
        android:text="@string/enable_vibration"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:padding="16dp" />
```

```

<Button
    android:id="@+id/chooseSoundButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/choose_notification_sound"
    app:layout_constraintTop_toBottomOf="@+id/vibrationSwitch"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:padding="16dp"
    app:layout_constraintHorizontal_bias="0.5" />

<Button
    android:id="@+id/chooseLedColorButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/choose_led_color"
    app:layout_constraintTop_toBottomOf="@+id/chooseSoundButton"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:padding="16dp"
    app:layout_constraintHorizontal_bias="0.5" />

<Button
    android:id="@+id/previewNotificationButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Preview Notification"
    app:layout_constraintTop_toBottomOf="@+id/chooseLedColorButton"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

java code:

MainActivity:

```

package com.example.la4q4;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    private final String CHANNEL_ID = "custom_channel_id";
    private final int NOTIFICATION_ID = 100;
    SharedPreferences sharedpreferences;

```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    sharedpreferences = getSharedPreferences("notification_settings",
MODE_PRIVATE);

    openSettingsButton = findViewById(R.id.openSettingsButton);
    previewButton = findViewById(R.id.previewButton);

    openSettingsButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this,
SettingsActivity.class);
            startActivity(intent);
        }
    });

    previewButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            triggerNotification();
        }
    });
}

createNotificationChannel();
}

private void triggerNotification() {
    boolean vibrationEnabled = sharedpreferences.getBoolean("vibration",
true);
    String soundUriString = sharedpreferences.getString("sound",
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION).toString());
    int ledColor = sharedpreferences.getInt("led_color", Color.BLUE);

    Uri soundUri = Uri.parse(soundUriString);

    Intent intent = new Intent(this, MainActivity.class);
    PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent,
PendingIntent.FLAG_IMMUTABLE);

    NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
        .setSmallIcon(R.drawable.notification)
        .setContentTitle("Custom Notification")
        .setContentText("This notification has custom settings")
        .setSound(soundUri)
        .setLights(ledColor, 1000, 1000)
        .setContentIntent(pendingIntent)
        .setAutoCancel(true);

    if (vibrationEnabled) {
        long[] vibrationPattern = {0, 500, 500, 500};
        builder.setVibrate(vibrationPattern);
    }

    NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(this);
```

```

        if (ActivityCompat.checkSelfPermission(this,
                android.Manifest.permission.POST_NOTIFICATIONS) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
        notificationManager.notify(NOTIFICATION_ID, builder.build());
    }

    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            String name = "Custom Channel";
            String description = "This channel is used for custom notifications";
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
                    name, importance);
            channel.setDescription(description);

            NotificationManager notificationManager =
                    getSystemService(NotificationManager.class);
            notificationManager.createNotificationChannel(channel);
        }
    }
}

```

SettingsActivity:

```

package com.example.la4q4;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Context;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.content.Intent;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.SwitchCompat;
import androidx.core.app.NotificationCompat;
import androidx.activity.result.ActivityResultLauncher;
import androidx.activity.result.contract.ActivityResultContracts;
import android.view.MenuItem;
import android.widget.Button;

import com.skydoves.colorpickerview.ColorEnvelope;
import com.skydoves.colorpickerview.ColorPickerDialog;
import com.skydoves.colorpickerview.listeners.ColorEnvelopeListener;

public class SettingsActivity extends AppCompatActivity {

    private Button selectSoundButton, selectColorButton, previewButton;
    private SwitchCompat vibrationSwitch;
    private Uri selectedSound;
    private int selectedLedColor;
    private boolean vibrationEnabled;

    private ActivityResultLauncher<Intent> ringtonePickerLauncher;

    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_settings);

    if (getSupportActionBar() != null) {
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }

    selectSoundButton = findViewById(R.id.chooseSoundButton);
    selectColorButton = findViewById(R.id.chooseLedColorButton);
    previewButton = findViewById(R.id.previewNotificationButton);
    vibrationSwitch = findViewById(R.id.vibrationSwitch);

    Sharedpreferences preferences =
getSharedpreferences("NotificationPrefs", MODE_PRIVATE);
    selectedSound = Uri.parse(preferences.getString("sound",
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION).toString()));
    selectedLedColor = preferences.getInt("ledColor", Color.RED);
    vibrationEnabled = preferences.getBoolean("vibration", true);

    vibrationSwitch.setChecked(vibrationEnabled);

    ringtonePickerLauncher = registerForActivityResult(new
ActivityResultContracts.StartActivityForResult(), result -> {
        if (result.getResultCode() == RESULT_OK && result.getData() != null)
{
            selectedSound =
result.getData().getParcelableExtra(RingtoneManager.EXTRA_RINGTONE_PICKED_URI);
            if (selectedSound != null) {
                Toast.makeText(SettingsActivity.this, "Sound selected",
Toast.LENGTH_SHORT).show();
            }
        }
    });
}

selectSoundButton.setOnClickListener(v -> {
    Intent intent = new Intent(RingtoneManager.ACTION_RINGTONE_PICKER);
    intent.putExtra(RingtoneManager.EXTRA_RINGTONE_TYPE,
RingtoneManager.TYPE_NOTIFICATION);
    intent.putExtra(RingtoneManager.EXTRA_RINGTONE_TITLE, "Select
Notification Sound");
    intent.putExtra(RingtoneManager.EXTRA_RINGTONE_EXISTING_URI,
selectedSound);
    ringtonePickerLauncher.launch(intent);
});

selectColorButton.setOnClickListener(v -> {
    new ColorPickerDialog.Builder(this)
        .setTitle("Pick LED Color")
        .setPreferenceName("LEDColorPicker")
        .setPositiveButton("Select", new ColorEnvelopeListener() {
            @Override
            public void onColorSelected(ColorEnvelope envelope,
boolean fromUser) {
                selectedLedColor = envelope.getColor();
                Toast.makeText(SettingsActivity.this, "Color
selected", Toast.LENGTH_SHORT).show();
            }
        })
        .setNegativeButton("Cancel", (dialogInterface, i) ->
dialogInterface.dismiss())
        .attachAlphaSlideBar(false)
        .attachBrightnessSlideBar(true)
});

```

```

        .show();
    });

    previewButton.setOnClickListener(v -> {
        vibrationEnabled = vibrationSwitch.isChecked();

        SharedPreferences.Editor editor = preferences.edit();
        editor.putString("sound", selectedSound.toString());
        editor.putInt("ledColor", selectedLedColor);
        editor.putBoolean("vibration", vibrationEnabled);
        editor.apply();

        showNotification();
    });
}

private void showNotification() {
    NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        NotificationChannel channel = new
NotificationChannel("default_channel", "Default Channel",
                NotificationManager.IMPORTANCE_DEFAULT);
        channel.setDescription("Notification Preview");
        notificationManager.createNotificationChannel(channel);
    }

    NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, "default_channel")
        .setSmallIcon(R.drawable.notification)
        .setContentTitle("Notification Preview")
        .setContentText("This is a preview of your custom
notification.")
        .setAutoCancel(true);

    if (vibrationEnabled) {
        long[] vibrationPattern = {0, 500, 1000};
        builder.setVibrate(vibrationPattern);
    }

    builder.setLights(selectedLedColor, 3000, 3000);

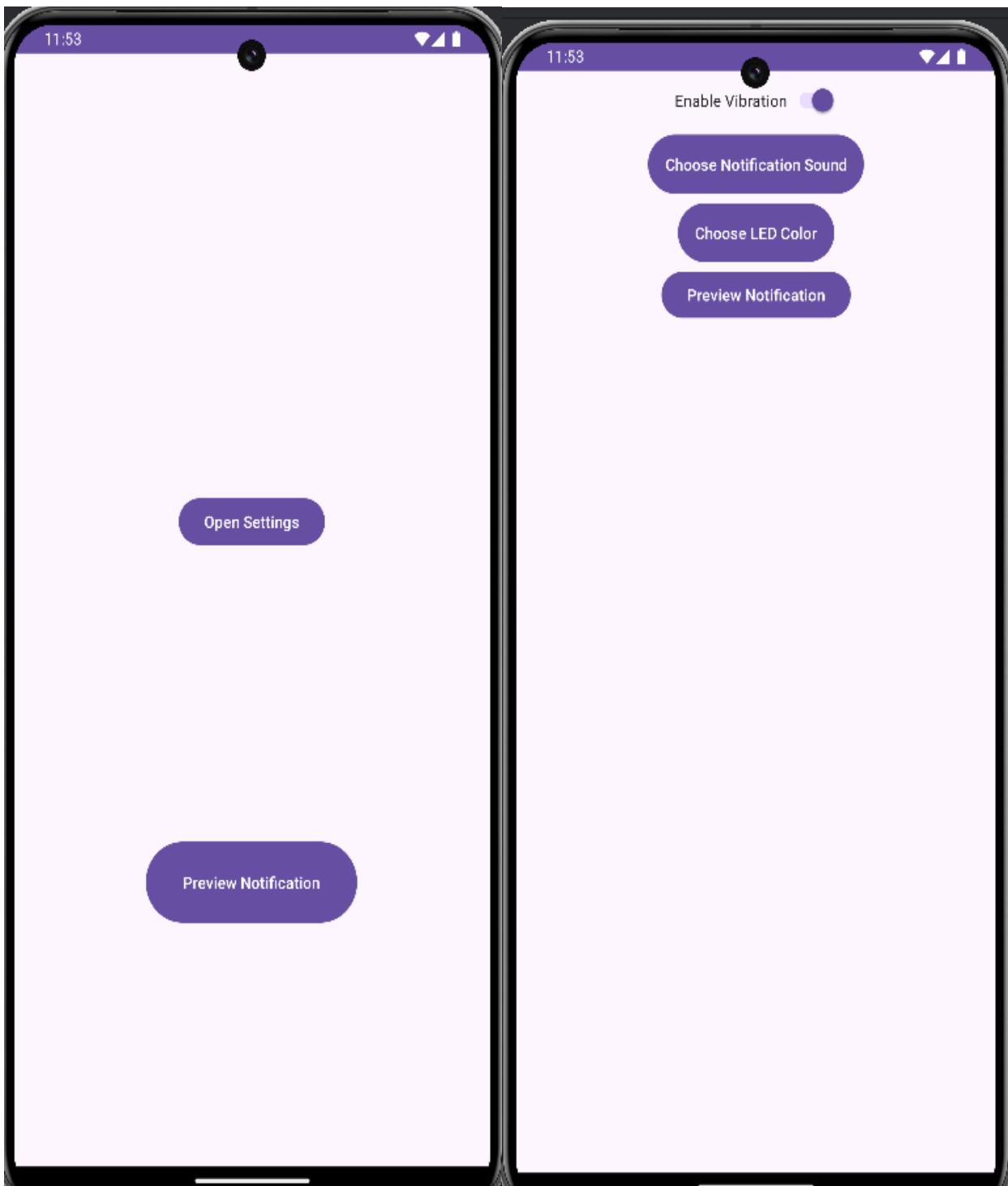
    if (selectedSound != null) {
        builder.setSound(selectedSound);
    }

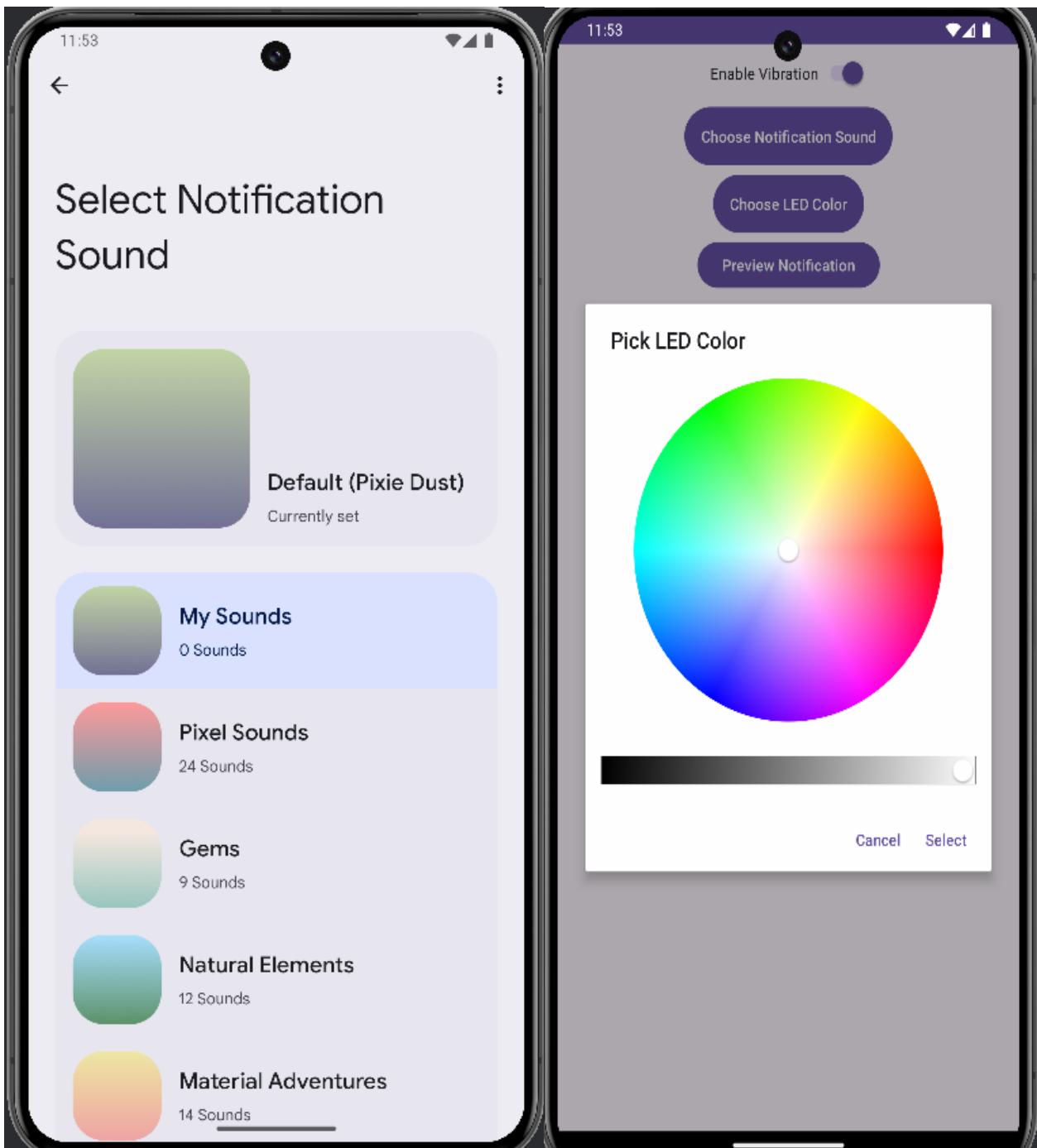
    notificationManager.notify(1, builder.build());
}

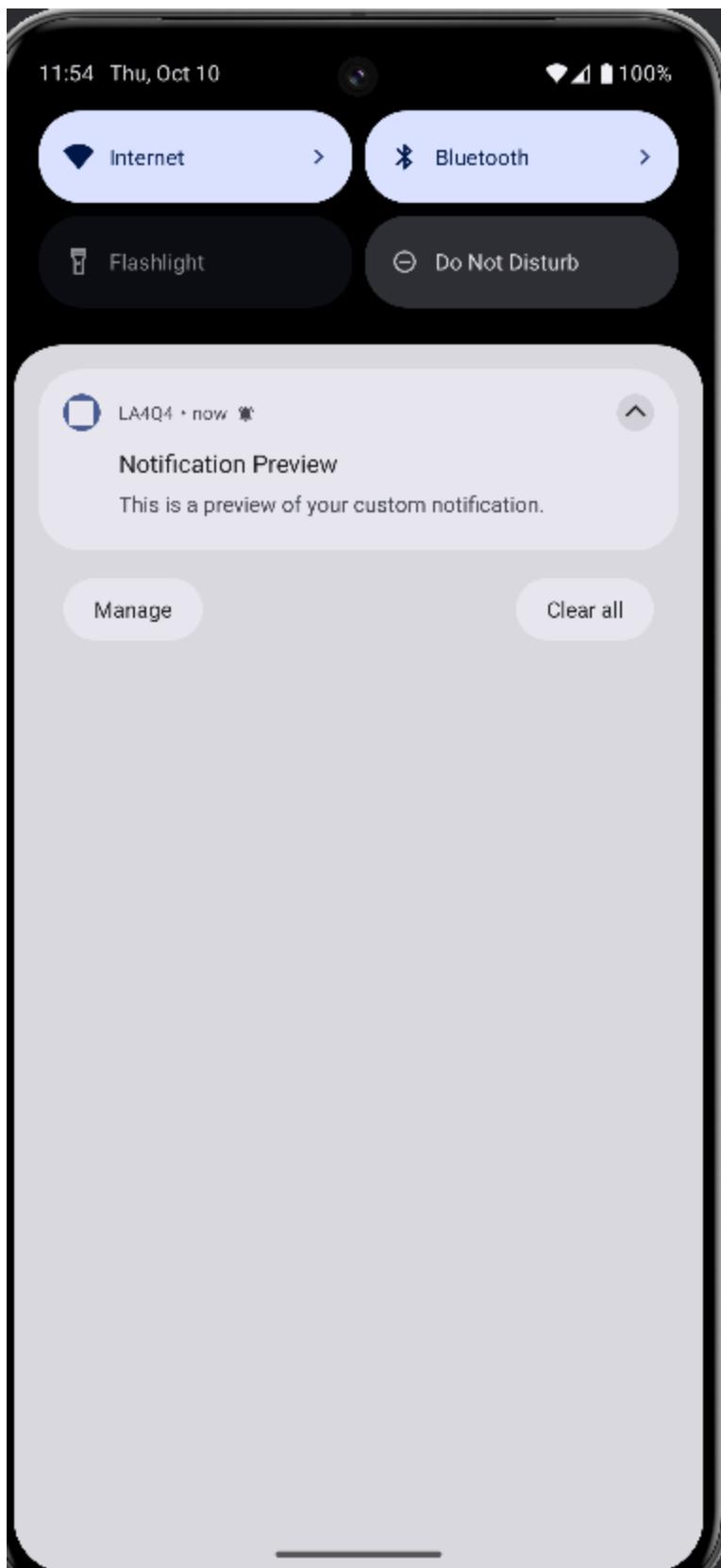
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if (item.getItemId() == android.R.id.home) {
        finish();
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

Output:







5. Create a notification that includes action buttons. For example, build a media player notification with "Play", and "Stop" buttons. Use the "Pause", NotificationCompat.Builder class to attach these actions and handle the corresponding intents when the user interacts with the notification.

Solution:

xml code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">

    <Button
        android:id="@+id/showNotificationButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Notification" />
</LinearLayout>
```

java code:

ActivityMain:

```
package com.example.la4q5;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    private static final String CHANNEL_ID = "media_channel";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button showNotificationButton =
findViewById(R.id.showNotificationButton);
        showNotificationButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showNotification(MainActivity.this);
            }
        });
    }

    public void showNotification(Context context) {

        Intent playIntent = new Intent(context, MediaPlayerReceiver.class);
        playIntent.setAction(MediaPlayerReceiver.ACTION_PLAY);
```

```

        PendingIntent playPendingIntent = PendingIntent.getBroadcast(context, 0,
playIntent, PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE);

        Intent stopIntent = new Intent(context, MediaPlayerReceiver.class);
        stopIntent.setAction(MediaPlayerReceiver.ACTION_STOP);
        PendingIntent stopPendingIntent = PendingIntent.getBroadcast(context, 1,
stopIntent, PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE);

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel channel = new NotificationChannel(
                CHANNEL_ID, "Media Player",
                NotificationManager.IMPORTANCE_LOW
            );
            NotificationManager notificationManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
            notificationManager.createNotificationChannel(channel);
        }

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(context, CHANNEL_ID)
            .setSmallIcon(R.drawable.media)
            .setContentTitle("Media Player")
            .setContentText("Now playing music")
            .setPriority(NotificationCompat.PRIORITY_LOW)
            .addAction(R.drawable.play, "Play", playPendingIntent)
            .addAction(R.drawable.stop, "Stop", stopPendingIntent)
            .setAutoCancel(true);

        NotificationManager notificationManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(1, builder.build());
    }
}

```

MediaPlayerReceiver:

```

package com.example.la4q5;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.MediaPlayer;
import android.widget.Toast;
import java.io.IOException;

public class MediaPlayerReceiver extends BroadcastReceiver {
    public static final String ACTION_PLAY =
"com.example.mediaplayernotification.ACTION_PLAY";
    public static final String ACTION_STOP =
"com.example.mediaplayernotification.ACTION_STOP";

    private MediaPlayer mediaPlayer;

    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent != null) {
            String action = intent.getAction();
            if (ACTION_PLAY.equals(action)) {
                String audioUrl =
"https://drive.google.com/uc?id=1l2QjyGJrunx_32vu243JdtO5tQxiZvMy";
                playAudio(context, audioUrl);
            } else if (ACTION_STOP.equals(action)) {

```

```

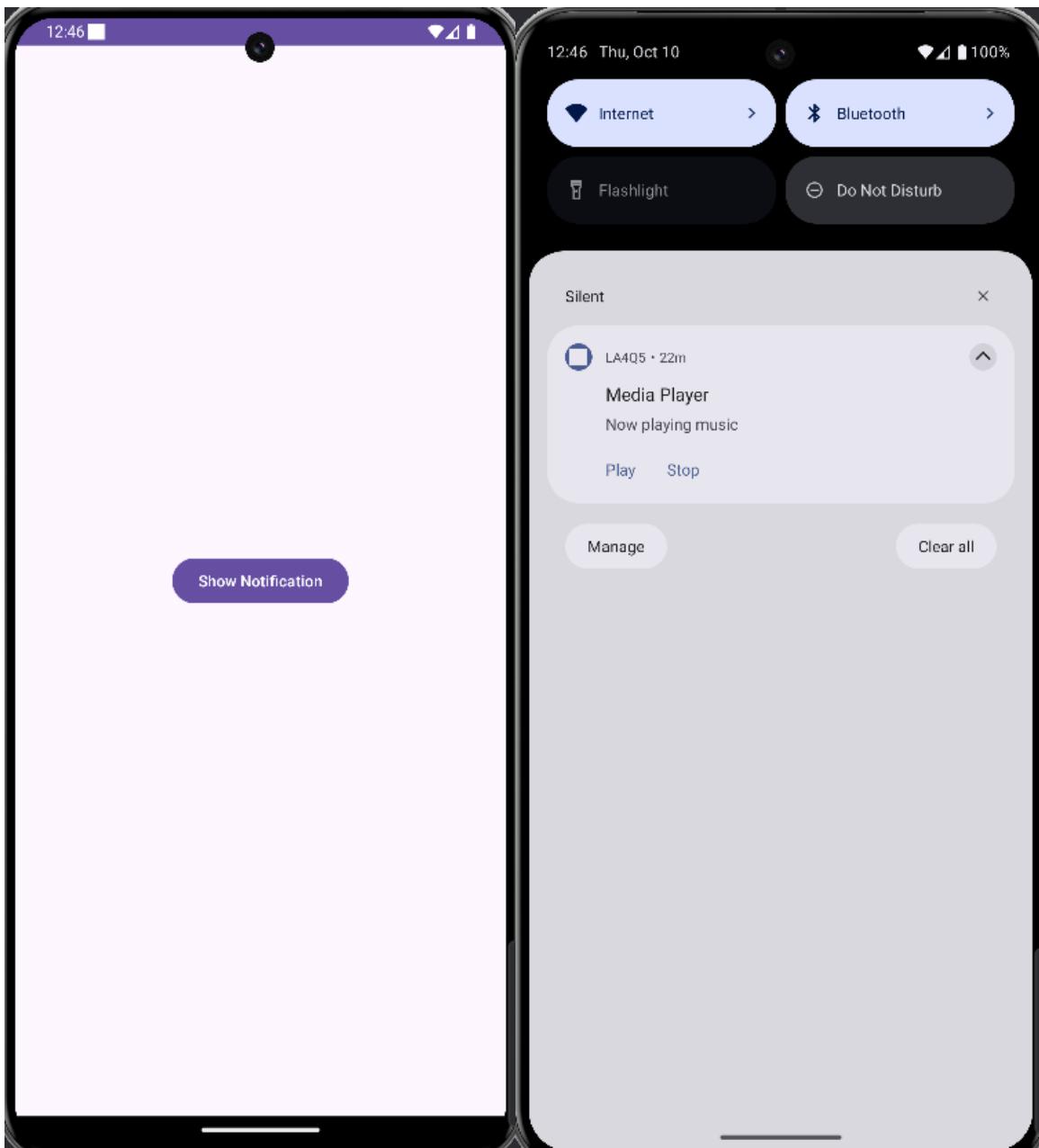
        stopAudio();
    }
}

private void playAudio(Context context, String url) {
    if (mediaPlayer == null) {
        mediaPlayer = new MediaPlayer();
        try {
            mediaPlayer.setDataSource(url);
            mediaPlayer.setOnPreparedListener(mp -> {
                if (!mp.isPlaying()) {
                    mp.start();
                    Toast.makeText(context, "Playback started",
Toast.LENGTH_SHORT).show();
                }
            });
            mediaPlayer.setOnCompletionListener(mp -> {
                stopAudio();
                Toast.makeText(context, "Playback completed",
Toast.LENGTH_SHORT).show();
            });
            mediaPlayer.prepareAsync();
        } catch (IOException e) {
            e.printStackTrace();
            Toast.makeText(context, "Error playing audio: " +
e.getMessage(), Toast.LENGTH_SHORT).show();
        }
    } else {
        Toast.makeText(context, "Audio is already playing",
Toast.LENGTH_SHORT).show();
    }
}

private void stopAudio() {
    if (mediaPlayer != null) {
        if (mediaPlayer.isPlaying()) {
            mediaPlayer.stop();
            Toast.makeText(null, "Playback stopped",
Toast.LENGTH_SHORT).show();
        }
        mediaPlayer.release();
        mediaPlayer = null;
    }
}

```

Output:



6. Develop an app that triggers should display a "Big large Picture Style" image notification. The notification expanded. Use when `NotificationCompat.BigPictureStyle` to implement the expanded notification and ensure it includes both a title and a summary text when collapsed.

Solution:

xml code:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/showNotificationButton"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Notification"
        android:layout_centerInParent="true" />
</RelativeLayout>
```

java code:

```
package com.example.la4q6;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Context;
import android.graphics.BitmapFactory;
import android.os.Build;
import android.os.Bundle;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

public class MainActivity extends AppCompatActivity {

    private static final String CHANNEL_ID = "big_picture_channel";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button showNotificationButton =
findViewByIId(R.id.showNotificationButton);
        showNotificationButton.setOnClickListener(v ->
showBigPictureNotification());

        createNotificationChannel();
    }

    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
                    "Big Picture Channel",
                    NotificationManager.IMPORTANCE_HIGH);
            channel.setDescription("Channel for big picture notifications");
            NotificationManager notificationManager =
getSystemService(NotificationManager.class);
            if (notificationManager != null) {
                notificationManager.createNotificationChannel(channel);
            }
        }
    }

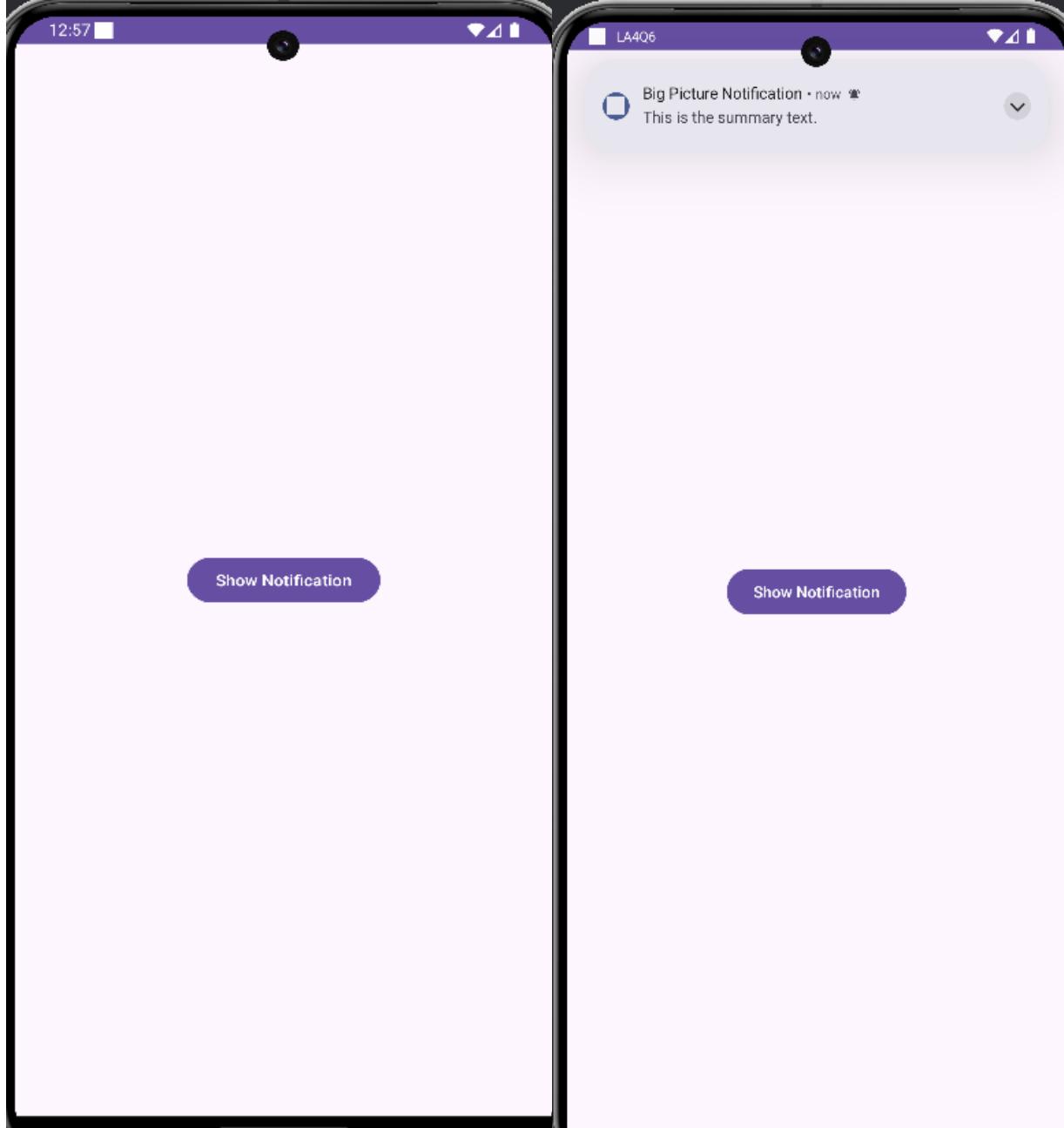
    private void showBigPictureNotification() {
        int imageResource = R.drawable.gfglogo;

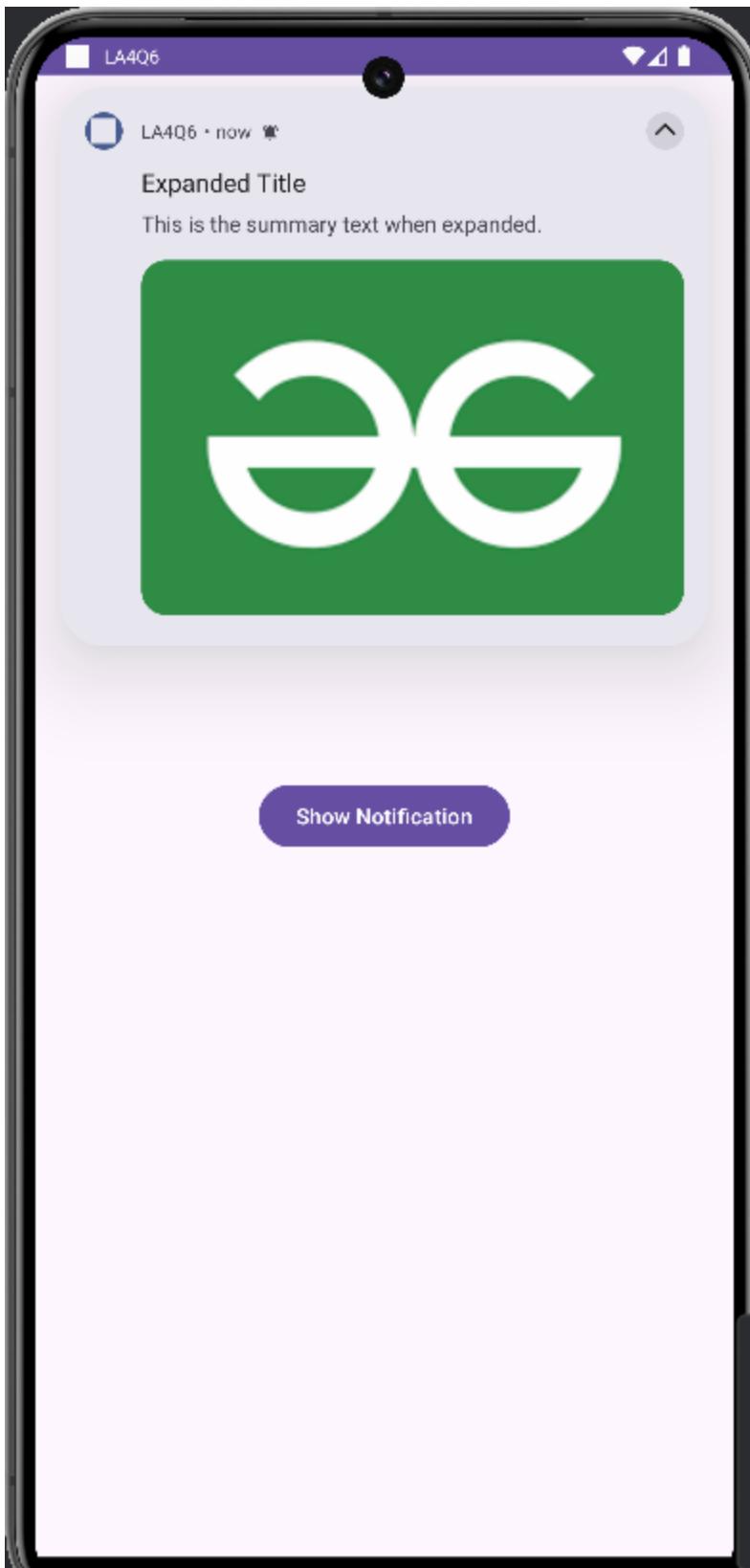
        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
            .setSmallIcon(R.drawable.notification)
            .setContentTitle("Big Picture Notification")
            .setContentText("This is the summary text.")
            .setPriority(NotificationCompat.PRIORITY_HIGH)
            ..setStyle(new NotificationCompat.BigPictureStyle()
                .bigPicture(BitmapFactory.decodeResource(getResources(),
imageResource)))
    }
}
```

```
        .setBigContentTitle("Expanded Title")
        .setSummaryText("This is the summary text when
expanded."))

    NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
    notificationManager.notify(1, builder.build());
}
}
```

Output:





7. Build an app that generates a heads-up notification (high-priority notification that pops up as an overlay). Set up the notification to appear when an urgent event

occurs, such as receiving an important message or a time-sensitive alert. Customize the notification to include an action, such as "Dismiss" or "Snooze".

Solution:

xml code:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/showNotificationButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Heads-Up Notification"
        android:layout_centerInParent="true" />
</RelativeLayout>
```

java code:

MainActivity:

```
package com.example.la4q7;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

public class MainActivity extends AppCompatActivity {

    private static final String CHANNEL_ID = "heads_up_channel";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button showNotificationButton =
findViewById(R.id.showNotificationButton);
        showNotificationButton.setOnClickListener(v ->
showHeadsUpNotification());

        createNotificationChannel();
    }

    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
                "Heads-Up Channel",
                NotificationManager.IMPORTANCE_HIGH);
            channel.setDescription("Channel for heads-up notifications");
            NotificationManager notificationManager =
getSystemService(NotificationManager.class);
            if (notificationManager != null) {
                notificationManager.createNotificationChannel(channel);
            }
        }
    }
}
```

```

    }

    private void showHeadsUpNotification() {
        Intent dismissIntent = new Intent(this, NotificationReceiver.class);
        dismissIntent.setAction(NotificationReceiver.ACTION_DISMISS);

        PendingIntent dismissPendingIntent = PendingIntent.getBroadcast(
            this,
            0,
            dismissIntent,
            PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE
        );

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
            .setSmallIcon(R.drawable.noti)
            .setContentTitle("Important Alert!")
            .setContentText("You have an urgent message.")
            .setPriority(NotificationCompat.PRIORITY_HIGH)
            .setAutoCancel(true)
            .addAction(R.drawable.dismiss, "Dismiss", dismissPendingIntent);

        NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(1, builder.build());
    }

}

```

NotificationReceiver:

```

package com.example.la4q7;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.widget.Toast;

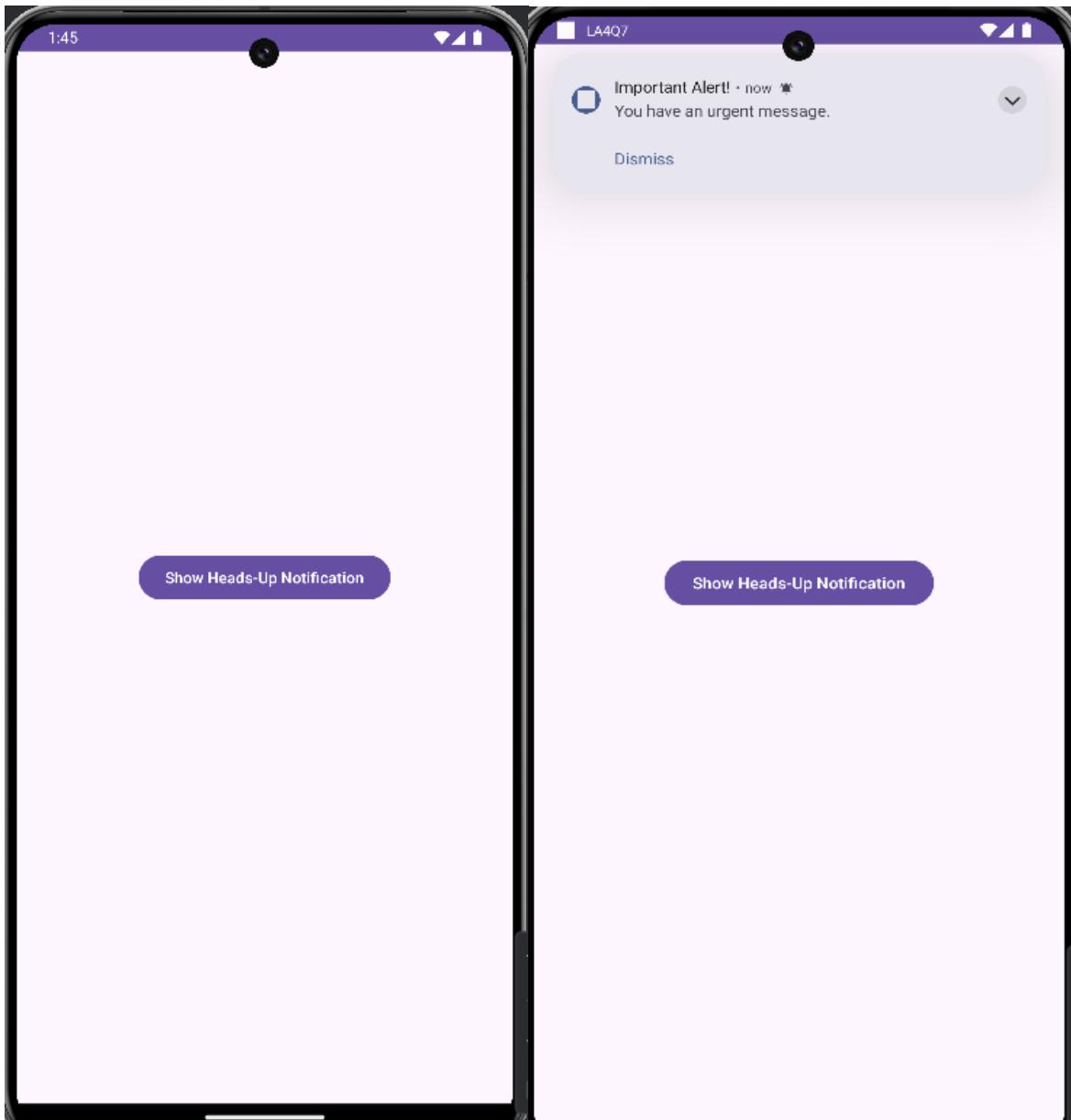
public class NotificationReceiver extends BroadcastReceiver {

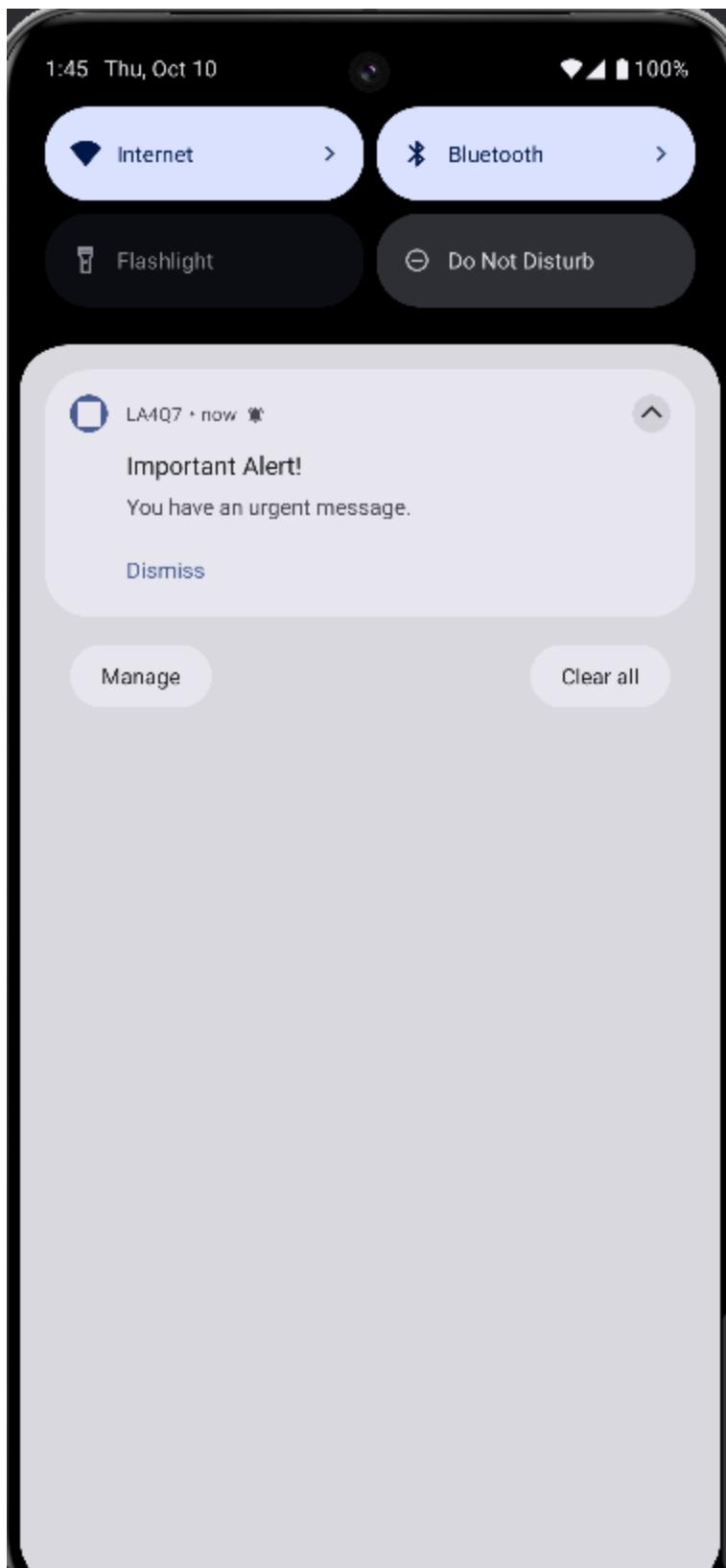
    public static final String ACTION_DISMISS =
"com.example.headsupnotification.ACTION_DISMISS";

    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent != null) {
            String action = intent.getAction();
            if (ACTION_DISMISS.equals(action)) {
                Toast.makeText(context, "Notification Dismissed",
Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

Output:





8. Develop an Android application that creates notification channels for different categories of notifications (e.g., "Messages", "Alerts", "Promotions"). Use the

NotificationChannel class to define channel properties like importance, sound, and vibration. Ensure notifications are issued under the appropriate channel, and allow the user to customize channel settings.

Solution:

xml code:

activity_main:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <Button
        android:id="@+id/button_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Message Notification" />

    <Button
        android:id="@+id/button_alert"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button_message"
        android:layout_marginTop="39dp"
        android:text="Send Alert Notification" />

    <Button
        android:id="@+id/button_promotion"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button_alert"
        android:layout_marginTop="34dp"
        android:text="Send Promotion Notification" />

    <Button
        android:id="@+id/button_settings"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button_promotion"
        android:layout_marginTop="175dp"
        android:text="Settings" />

</RelativeLayout>
```

activity_settings:

```
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SettingsActivity">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:layout_marginTop="?attr/actionBarSize">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Notification Settings"
        android:textSize="24sp"
        android:layout_marginBottom="16dp" />

    <Switch
        android:id="@+id/notificationsSwitch"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enable Notifications" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Notification Importance" />

    <Spinner
        android:id="@+id/importanceSpinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Notification Sound" />

    <Button
        android:id="@+id/selectSoundButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Select Sound" />

    <Button
        android:id="@+id/saveButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:text="Save Settings" />
</LinearLayout>

</androidx.coordinatorlayout.widget.CoordinatorLayout>

```

java code:

MainActivity:

```

package com.example.la4q8;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

```

```
import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;
import android.widget.Button;
import android.content.Intent;

public class MainActivity extends AppCompatActivity {

    private static final String CHANNEL_MESSAGES_ID = "messages_channel";
    private static final String CHANNEL_ALERTS_ID = "alerts_channel";
    private static final String CHANNEL_PROMOTIONS_ID = "promotions_channel";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        createNotificationChannels();

        Button messageButton = findViewById(R.id.button_message);
        messageButton.setOnClickListener(v -> sendNotification("Messages", "You have a new message!", CHANNEL_MESSAGES_ID));

        Button alertButton = findViewById(R.id.button_alert);
        alertButton.setOnClickListener(v -> sendNotification("Alerts", "This is an important alert!", CHANNEL_ALERTS_ID));

        Button promotionButton = findViewById(R.id.button_promotion);
        promotionButton.setOnClickListener(v -> sendNotification("Promotions", "Check out our latest promotions!", CHANNEL_PROMOTIONS_ID));
        Button settingsButton = findViewById(R.id.button_settings);
        settingsButton.setOnClickListener(v -> {
            Intent intent = new Intent(MainActivity.this,
SettingsActivity.class);
            startActivity(intent);
        });
    }

    private void createNotificationChannels() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel messagesChannel = new NotificationChannel(
                CHANNEL_MESSAGES_ID,
                "Messages",
                NotificationManager.IMPORTANCE_HIGH
            );
            messagesChannel.setDescription("Notifications for messages");
            messagesChannel.setVibrationPattern(new long[]{0, 1000, 500, 1000});
            messagesChannel.enableVibration(true);

            NotificationChannel alertsChannel = new NotificationChannel(
                CHANNEL_ALERTS_ID,
                "Alerts",
                NotificationManager.IMPORTANCE_HIGH
            );
            alertsChannel.setDescription("Notifications for alerts");
            alertsChannel.setVibrationPattern(new long[]{0, 500, 250, 500});
            alertsChannel.enableVibration(true);

            NotificationChannel promotionsChannel = new NotificationChannel(
                CHANNEL_PROMOTIONS_ID,
                "Promotions",
                NotificationManager.IMPORTANCE_LOW
            );
        }
    }
}
```

```
    );
    promotionsChannel.setDescription("Notifications for promotions");
    promotionsChannel.setVibrationPattern(new long[]{0, 300, 100, 300});
    promotionsChannel.enableVibration(false);

    NotificationManager notificationManager =
getSystemService(NotificationManager.class);
    if (notificationManager != null) {
        notificationManager.createNotificationChannel(messagesChannel);
        notificationManager.createNotificationChannel(alertsChannel);
notificationManager.createNotificationChannel(promotionsChannel);
    }
}

private void sendNotification(String title, String message, String
channelId) {
    NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, channelId)
        .setSmallIcon(R.drawable.noti)
        .setContentTitle(title)
        .setContentText(message)
        .setPriority(NotificationCompat.PRIORITY_HIGH)
        .setAutoCancel(true);

    NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(this);
    if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {
        return;
    }
    notificationManager.notify((int) System.currentTimeMillis(),
builder.build());
}

private void createNotificationChannel() {
    SharedPreferences preferences = getSharedPreferences("AppPreferences",
MODE_PRIVATE);
    boolean notificationsEnabled =
preferences.getBoolean("notifications_enabled", true);
    String importanceLevel =
preferences.getString("notification_importance", "Default");
    String soundUriString = preferences.getString("notification_sound",
null);
    Uri soundUri = soundUriString != null ? Uri.parse(soundUriString) :
null;

    if (notificationsEnabled) {
        int importance = NotificationManager.IMPORTANCE_DEFAULT;
        switch (importanceLevel) {
            case "High":
                importance = NotificationManager.IMPORTANCE_HIGH;
                break;
            case "Low":
                importance = NotificationManager.IMPORTANCE_LOW;
                break;
            case "Max":
                importance = NotificationManager.IMPORTANCE_MAX;
                break;
            case "Min":
                importance = NotificationManager.IMPORTANCE_MIN;
                break;
        }
    }
}
```

```

        NotificationChannel channel = null;
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            channel = new NotificationChannel("YourChannelId", "Your Channel
Name", importance);
        }
        if (soundUri != null) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                channel.setSound(soundUri, null);
            }
        }
        NotificationManager notificationManager =
getSystemService(NotificationManager.class);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            notificationManager.createNotificationChannel(channel);
        }
    }
}
}

```

SettingsActivity:

```

package com.example.la4q8;

import android.content.Intent;
import android.content.SharedPreferences;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Switch;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

public class SettingsActivity extends AppCompatActivity {

    private Switch notificationsSwitch;
    private Spinner importanceSpinner;
    private Button selectSoundButton;
    private Button saveButton;
    private Uri notificationSound;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_settings);

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        notificationsSwitch = findViewById(R.id.notificationsSwitch);
        importanceSpinner = findViewById(R.id.importanceSpinner);
        selectSoundButton = findViewById(R.id.selectSoundButton);
        saveButton = findViewById(R.id.saveButton);

        setupImportanceSpinner();

        selectSoundButton.setOnClickListener(v -> selectNotificationSound());
        saveButton.setOnClickListener(v -> saveSettings());
    }
}

```

```

    }

    private void setupImportanceSpinner() {
        ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(this,
            R.array.importance_levels,
        android.R.layout.simple_spinner_item);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        importanceSpinner.setAdapter(adapter);
    }

    private void selectNotificationSound() {
        Intent intent = new Intent(RingtoneManager.ACTION_RINGTONE_PICKER);
        intent.putExtra(RingtoneManager.EXTRA_RINGTONE_TYPE,
RingtoneManager.TYPE_NOTIFICATION);
        intent.putExtra(RingtoneManager.EXTRA_RINGTONE_TITLE, "Select
Notification Sound");
        intent.putExtra(RingtoneManager.EXTRA_RINGTONE_EXISTING_URI,
notificationSound);
        startActivityForResult(intent, 1);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        if (requestCode == 1 && resultCode == RESULT_OK) {
            notificationSound =
data.getParcelableExtra(RingtoneManager.EXTRA_RINGTONE_PICKED_URI);
        }
        super.onActivityResult(requestCode, resultCode, data);
    }

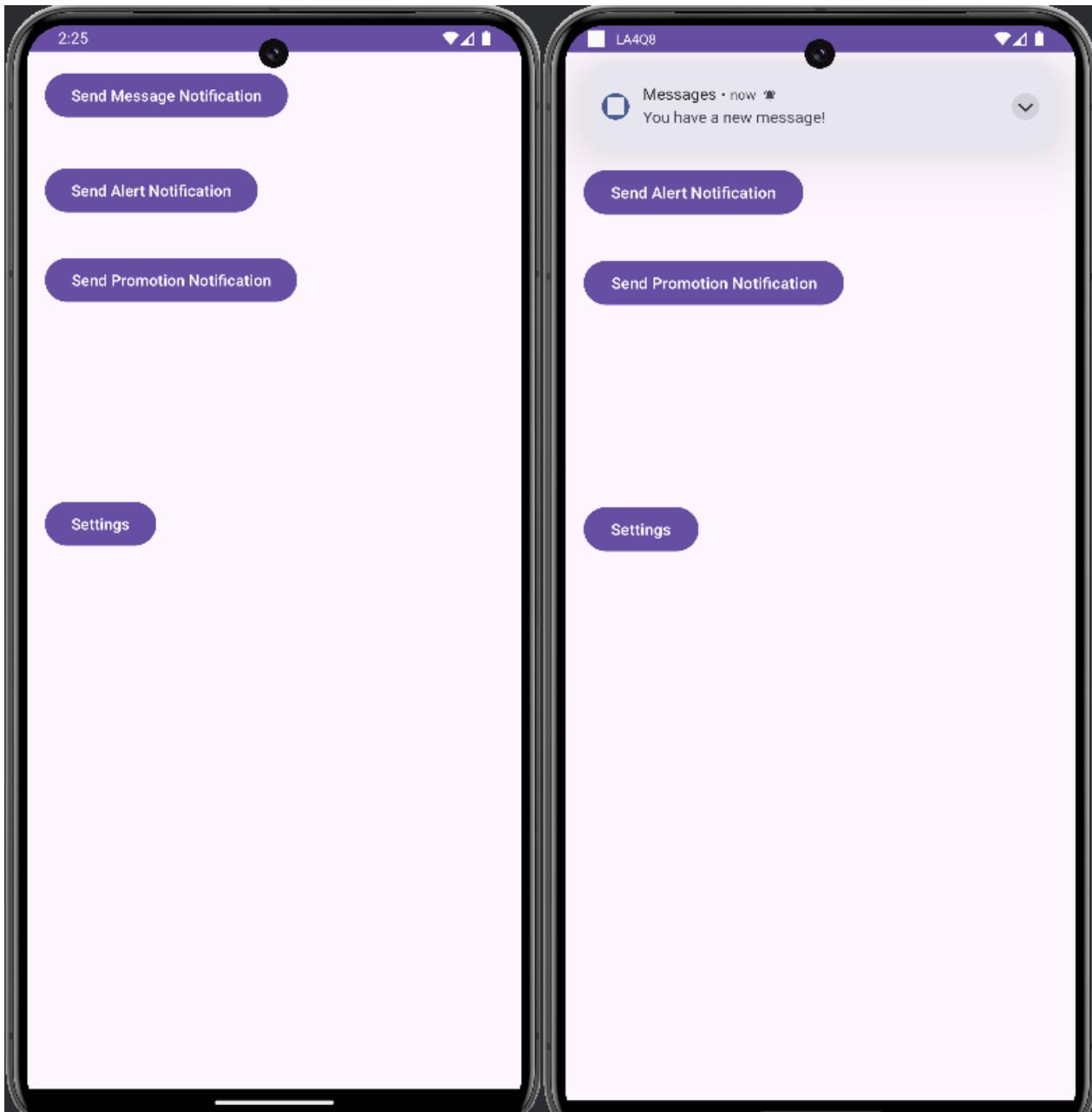
    private void saveSettings() {
        SharedPreferences preferences = getSharedPreferences("AppPreferences",
        MODE_PRIVATE);
        SharedPreferences.Editor editor = preferences.edit();

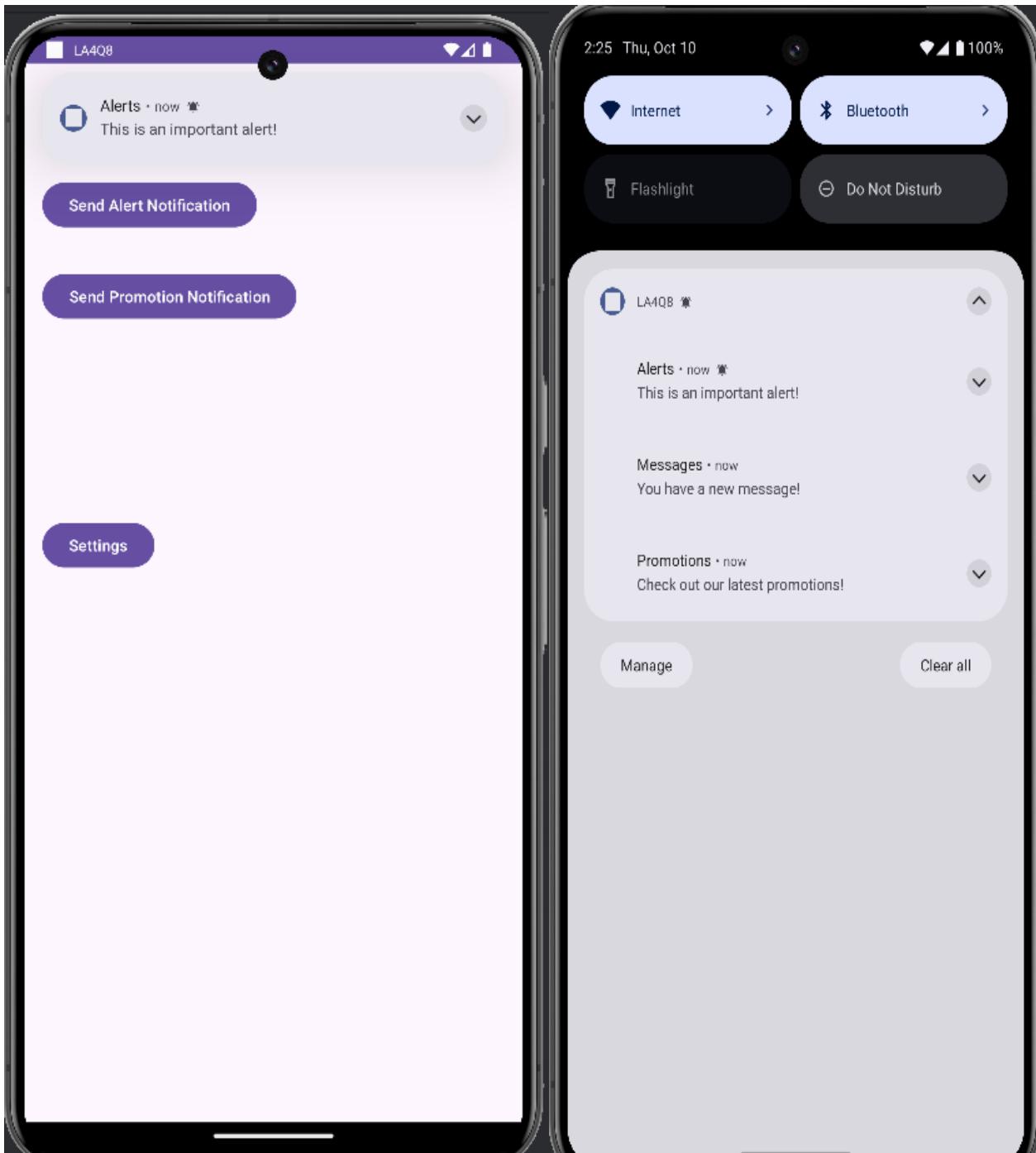
        editor.putBoolean("notifications_enabled",
notificationsSwitch.isChecked());
        editor.putString("notification_importance",
importanceSpinner.getSelectedItem().toString());
        editor.putString("notification_sound", notificationSound != null ?
notificationSound.toString() : null);

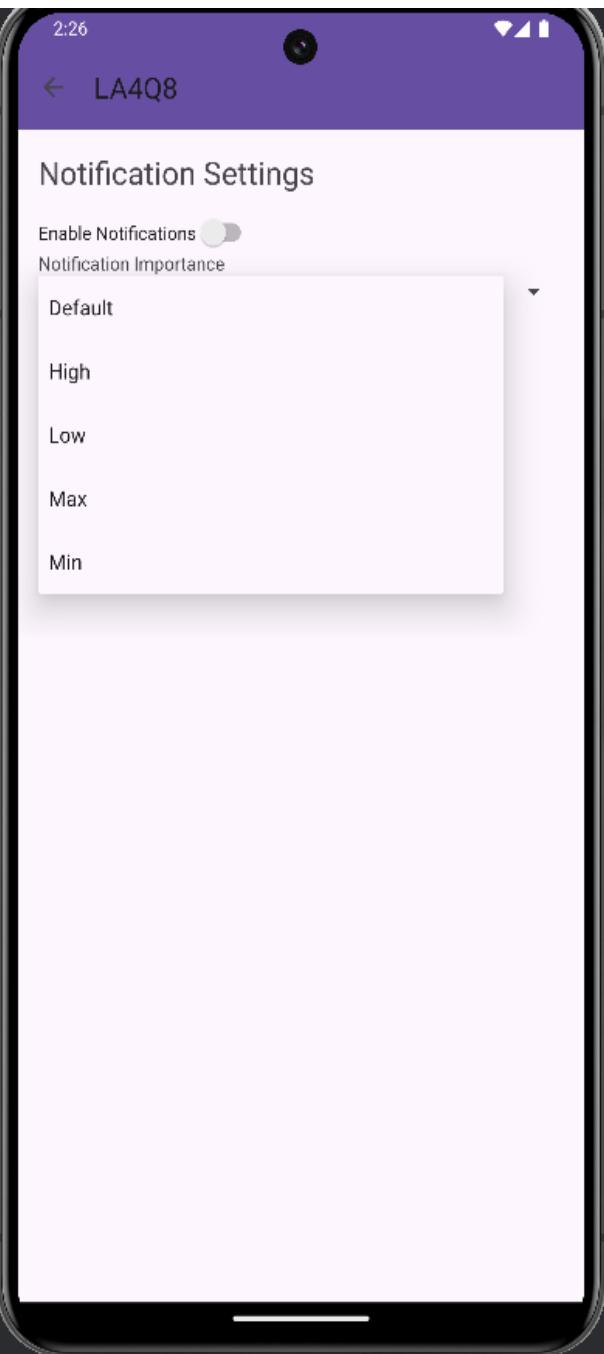
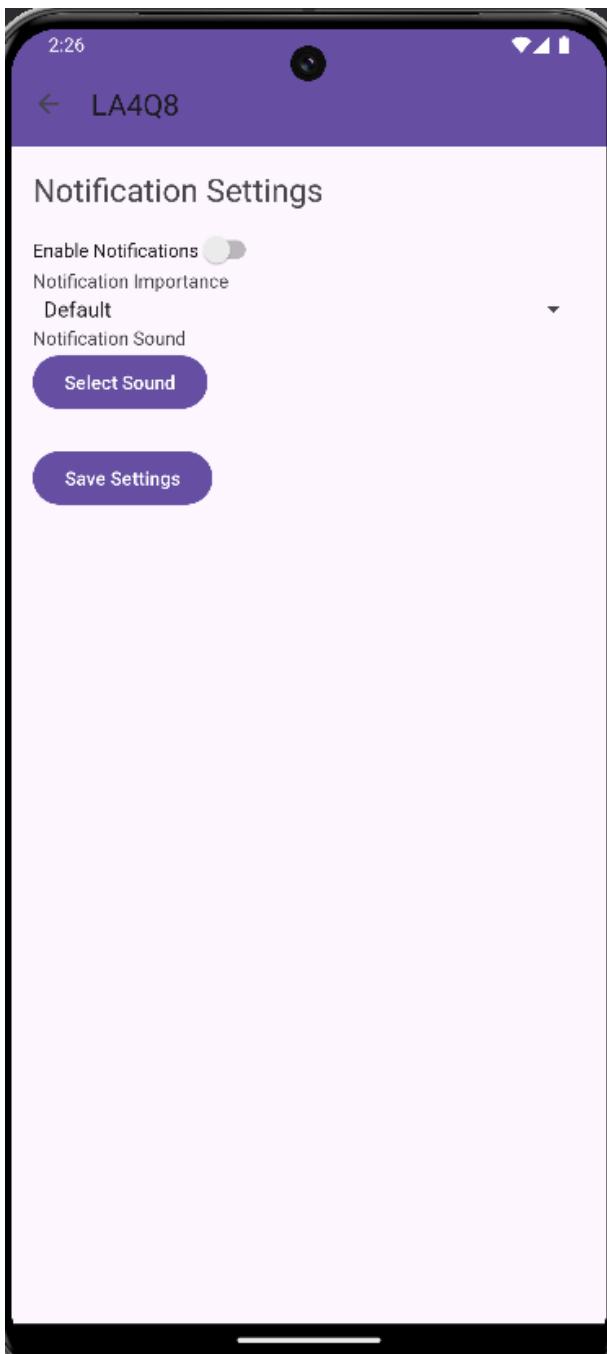
        editor.apply();
        finish();
    }
}

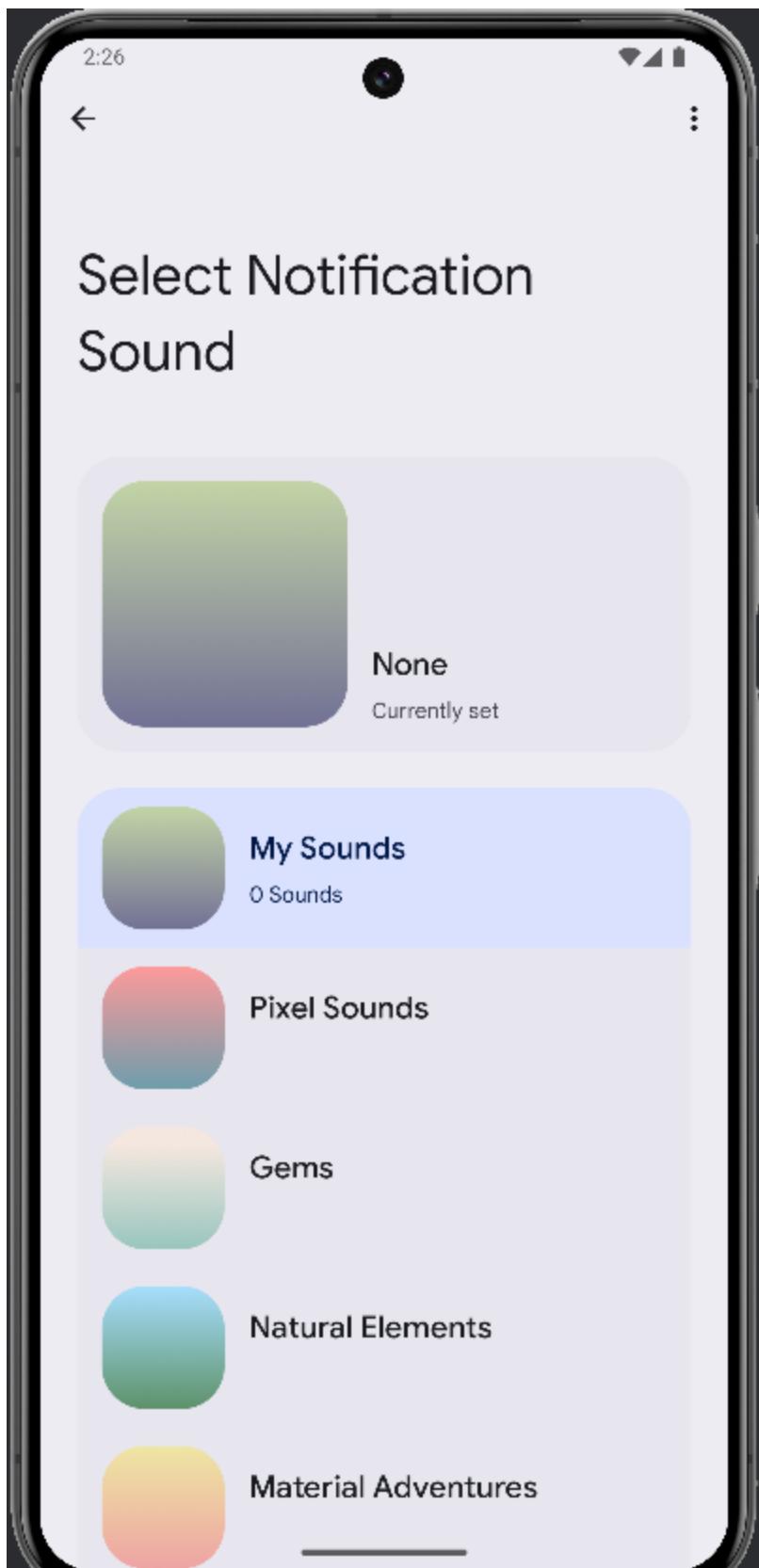
```

Output:









9. Create an application that issues multiple notifications and groups them into a single expandable notification. Use `NotificationCompat.Builder` and

`NotificationCompat.InboxStyle` to group notifications, such as showing a list of recent messages in a messaging app. Implement functionality to expand and collapse the group.

Solution:

xml code:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/send_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Notification"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

java code:

```
package com.example.la4q9;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

public class MainActivity extends AppCompatActivity {

    private static final String CHANNEL_ID = "message_channel";
    private int notificationCount = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button sendButton = findViewById(R.id.send_button);
        createNotificationChannel();

        sendButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                sendGroupedNotification("Message from User", "Hello, this is message number " + (notificationCount + 1));
                notificationCount++;
            }
        });
    }

    private void sendGroupedNotification(String title, String message) {
        NotificationManager notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
```

```

        Intent intent = new Intent(this, MainActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);

        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent,
PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE);

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
        .setSmallIcon(R.drawable.noti)
        .setContentTitle(title)
        .setContentText(message)
        .setContentIntent(pendingIntent)
        .setAutoCancel(true)
        .setGroup("messages_group")
        .setGroupSummary(false);

        notificationManager.notify(notificationCount, builder.build());

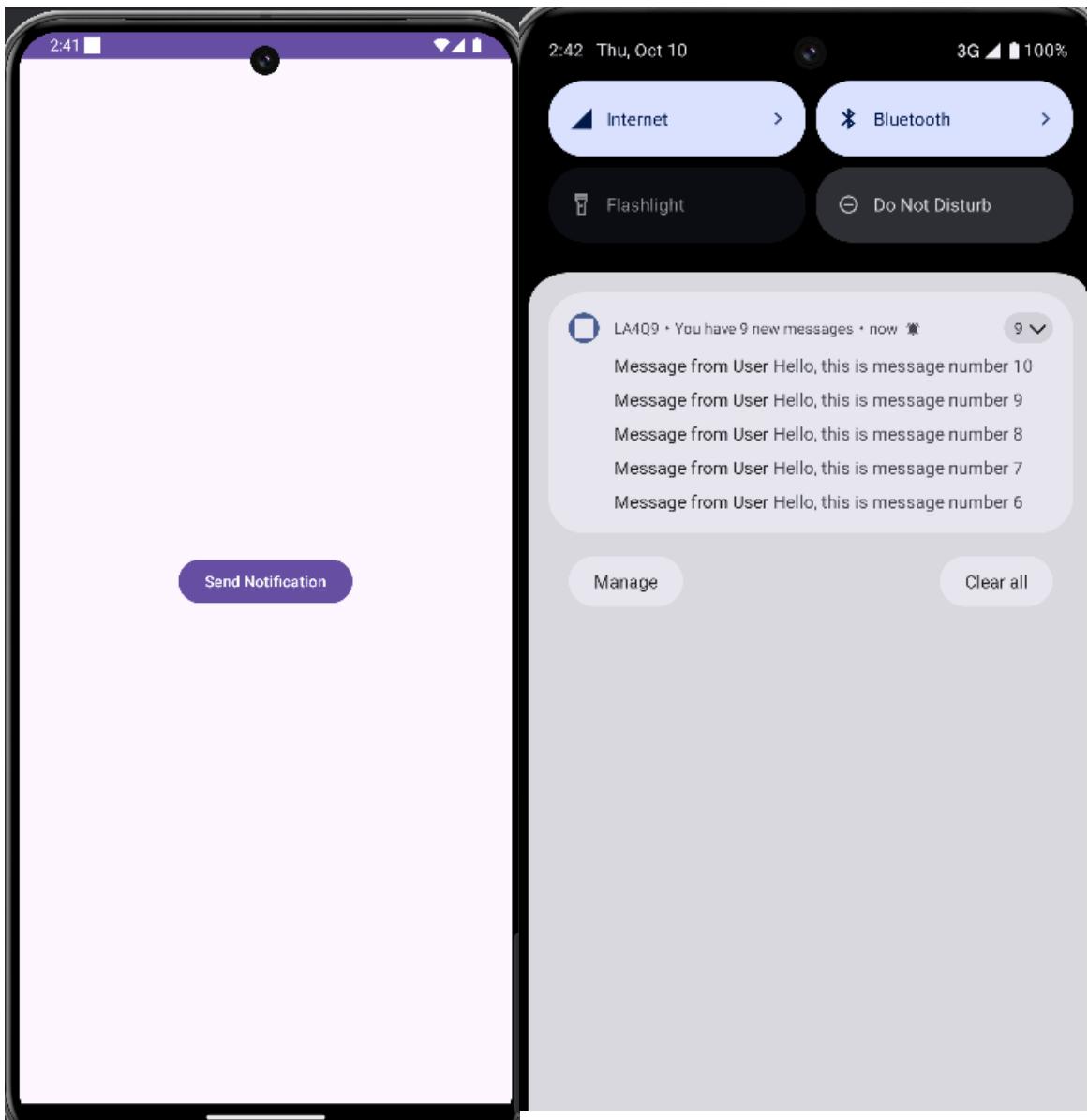
        NotificationCompat.Builder summaryBuilder = new
NotificationCompat.Builder(this, CHANNEL_ID)
        .setContentTitle("New Messages")
        .setContentText("You have " + notificationCount + " new
messages")
        .setSmallIcon(R.drawable.noti)
        ..setStyle(new NotificationCompat.InboxStyle()
            .addLine("Message from User: Hello, this is message
number " + notificationCount)
            .setBigContentTitle("New Messages")
            .setSummaryText("You have " + notificationCount + " new
messages"))
        .setGroup("messages_group")
        .setGroupSummary(true)
        .setContentIntent(pendingIntent);

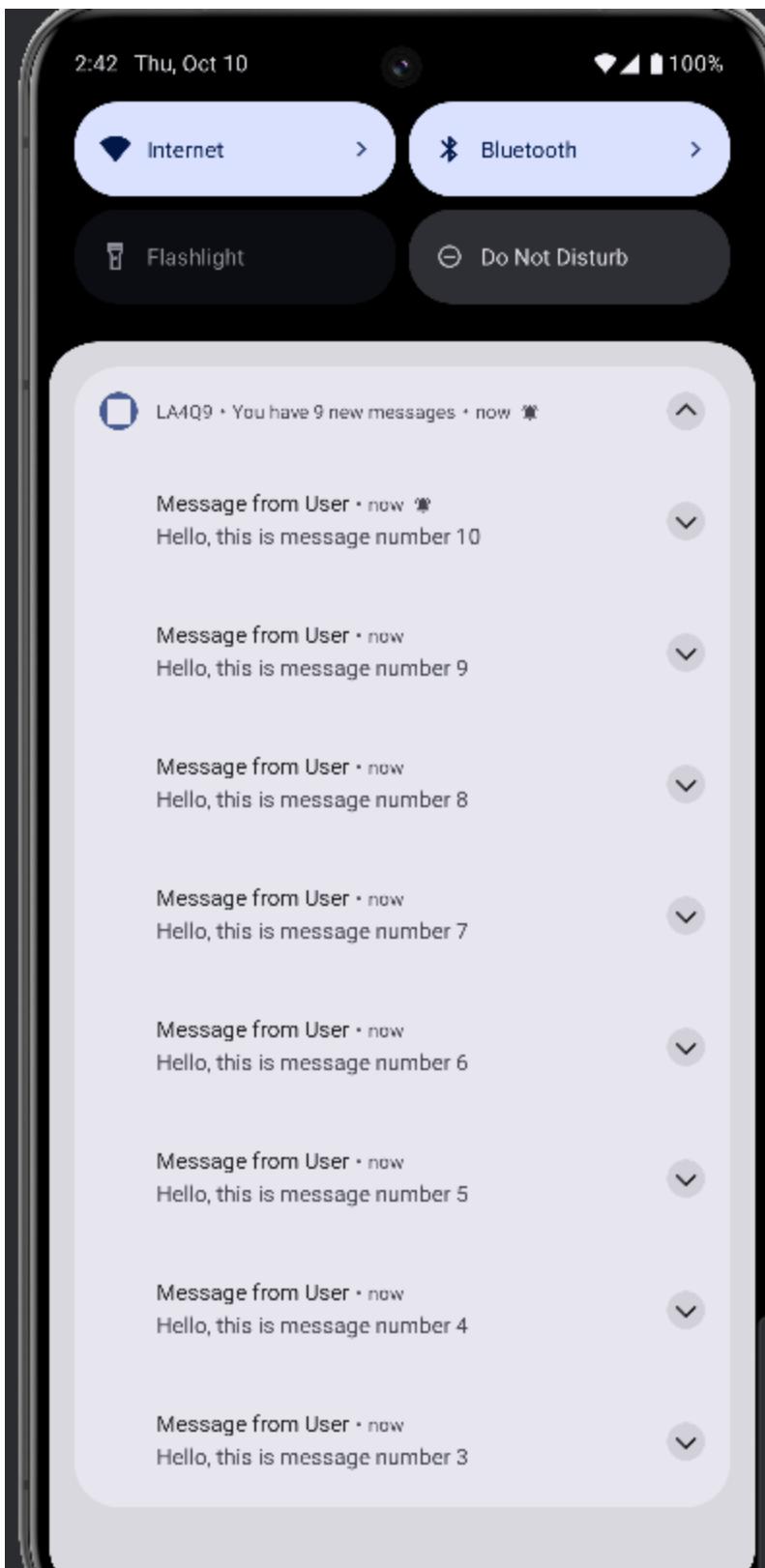
        notificationManager.notify(0, summaryBuilder.build());
    }

    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel channel = new NotificationChannel(
                CHANNEL_ID,
                "Message Notifications",
                NotificationManager.IMPORTANCE_DEFAULT);
            NotificationManager manager =
getSystemService(NotificationManager.class);
            manager.createNotificationChannel(channel);
        }
    }
}

```

Output:





10. Design an application that schedules and triggers notifications at a specific time or interval (e.g., daily reminders). Use AlarmManager or WorkManager to schedule the notifications, and issue them using NotificationCompat.Builder. Ensure that notifications are triggered even when the app is in the background or closed.

Solution:

xml code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/select_time"
        android:textSize="18sp"
        android:layout_marginBottom="16dp"/>

    <TimePicker
        android:id="@+id/timePicker"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:timePickerMode="spinner"/>

    <Button
        android:id="@+id/scheduleButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/schedule_notification"
        android:layout_marginTop="16dp"/>

</LinearLayout>
```

java code:

MainActivity:

```
package com.example.la4q10;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TimePicker;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.work.Data;
import androidx.work.ExistingPeriodicWorkPolicy;
import androidx.work.PeriodicWorkRequest;
import androidx.work.WorkManager;

import java.util.Calendar;
import java.util.concurrent.TimeUnit;

public class MainActivity extends AppCompatActivity {
    public static final String CHANNEL_ID = "reminder_channel";
```

```
public static final int NOTIFICATION_ID = 1;
public static final String NOTIFICATION_HOUR = "notification_hour";
public static final String NOTIFICATION_MINUTE = "notification_minute";

private TimePicker timePicker;
private Button scheduleButton;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    timePicker = findViewById(R.id.timePicker);
    scheduleButton = findViewById(R.id.scheduleButton);
    timePicker.setIs24HourView(true);

    createNotificationChannel();

    scheduleButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            scheduleNotification();
        }
    });
}

private void createNotificationChannel() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        CharSequence name = getString(R.string.channel_name);
        String description = getString(R.string.channel_description);
        int importance = NotificationManager.IMPORTANCE_DEFAULT;
        NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
name, importance);
        channel.setDescription(description);

        NotificationManager notificationManager =
getSystemService(NotificationManager.class);
        if (notificationManager != null) {
            notificationManager.createNotificationChannel(channel);
        }
    }
}

private void scheduleNotification() {
    int selectedHour = timePicker.getHour();
    int selectedMinute = timePicker.getMinute();

    Data inputData = new Data.Builder()
        .putInt(NOTIFICATION_HOUR, selectedHour)
        .putInt(NOTIFICATION_MINUTE, selectedMinute)
        .build();

    Calendar calendar = Calendar.getInstance();
    Calendar selectedTime = Calendar.getInstance();
    selectedTime.set(Calendar.HOUR_OF_DAY, selectedHour);
    selectedTime.set(Calendar.MINUTE, selectedMinute);
    selectedTime.set(Calendar.SECOND, 0);

    if (selectedTime.before(calendar)) {
        selectedTime.add(Calendar.DAY_OF_MONTH, 1);
    }
}
```

```

        long initialDelay = selectedTime.getTimeInMillis() -
calendar.getTimeInMillis();

        PeriodicWorkRequest notificationWorkRequest =
                new PeriodicWorkRequest.Builder(NotificationWorker.class, 24,
TimeUnit.HOURS)
                        .setInputData(inputData)
                        .setInitialDelay(initialDelay, TimeUnit.MILLISECONDS)
                        .build();

WorkManager.getInstance(getApplicationContext()).enqueueUniquePeriodicWork(
        "dailyReminder",
        ExistingPeriodicWorkPolicy.REPLACE,
        notificationWorkRequest
);

        String message = String.format("Notification scheduled for %02d:%02d
daily",
                selectedHour, selectedMinute);
        Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
    }
}

```

NotificationWorker:

```

package com.example.la4q10;

import android.app.NotificationManager;
import android.content.Context;

import androidx.annotation.NonNull;
import androidx.core.app.NotificationCompat;
import androidx.work.Worker;
import androidx.work.WorkerParameters;
import androidx.work.Data;

public class NotificationWorker extends Worker {
    public NotificationWorker(@NonNull Context context, @NonNull WorkerParameters params) {
        super(context, params);
    }

    @NonNull
    @Override
    public Result doWork() {
        Data inputData = getInputData();
        int hour = inputData.getInt(MainActivity.NOTIFICATION_HOUR, -1);
        int minute = inputData.getInt(MainActivity.NOTIFICATION_MINUTE, -1);

        if (hour != -1 && minute != -1) {
            showNotification(hour, minute);
        }

        return Result.success();
    }

    private void showNotification(int hour, int minute) {
        Context context = getApplicationContext();
        if (context == null) return;

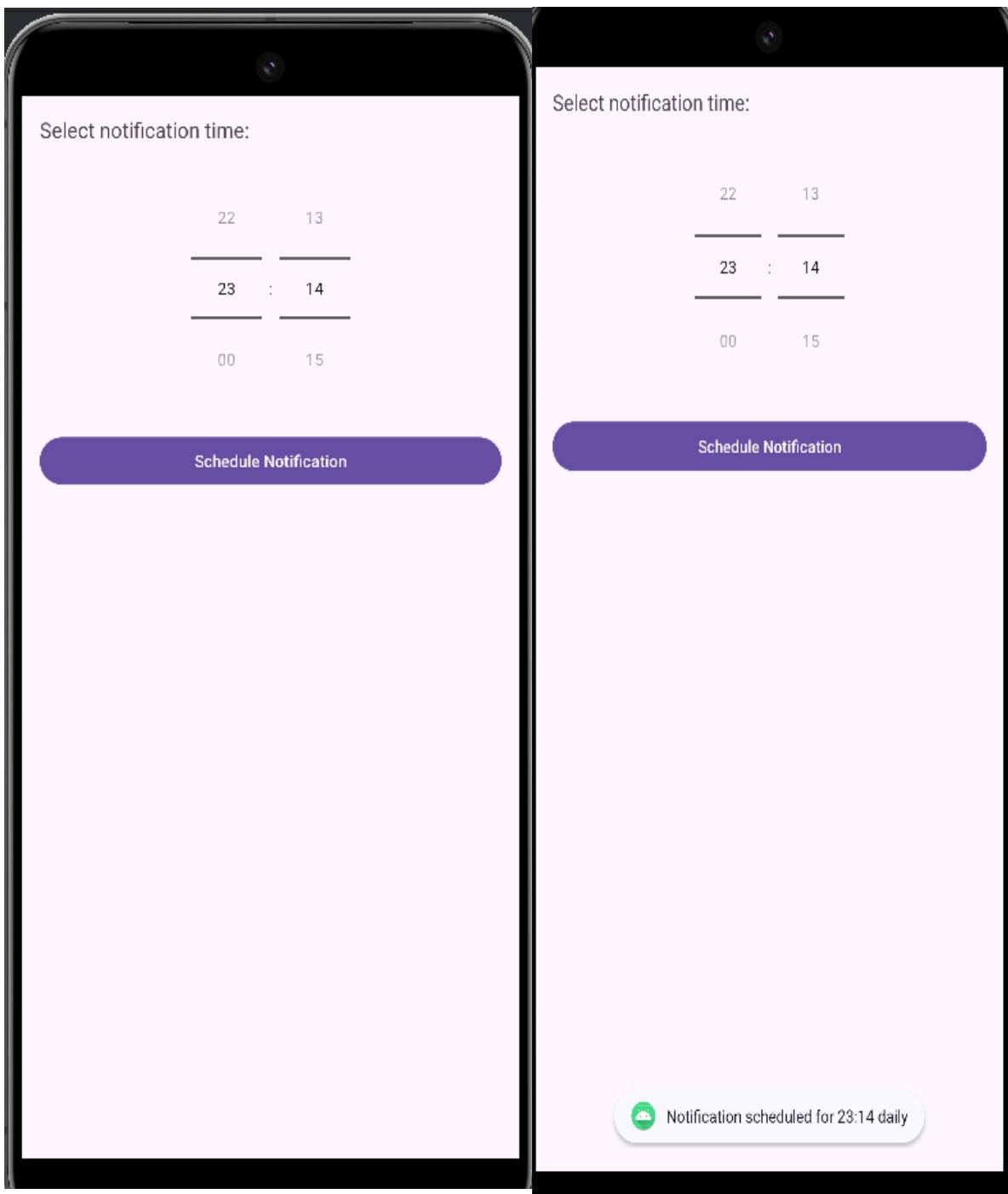
        String notificationText = String.format("It's %02d:%02d - Time for your
daily reminder!", hour, minute);
    }
}

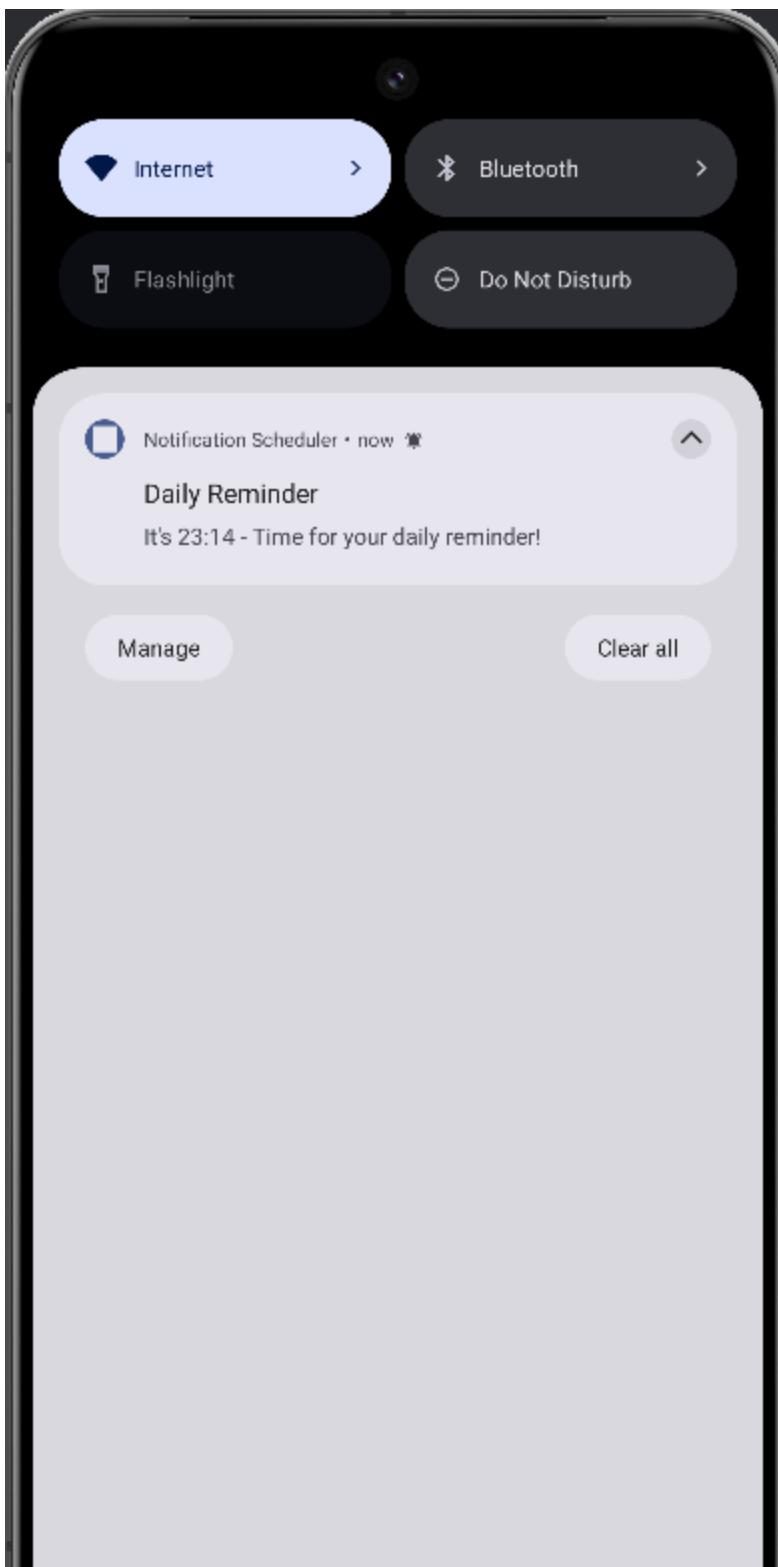
```

```
NotificationCompat.Builder builder = new
NotificationCompat.Builder(context, MainActivity.CHANNEL_ID)
    .setSmallIcon(R.drawable.noti)
    .setContentTitle("Daily Reminder")
    .setContentText(notificationText)
    .setPriority(NotificationCompat.PRIORITY_DEFAULT)
    .setAutoCancel(true);

NotificationManager notificationManager =
    (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);
if (notificationManager != null) {
    notificationManager.notify(MainActivity.NOTIFICATION_ID,
builder.build());
}
}
```

Output:







Pimpri Chinchwad Education Trust's Pimpri Chinchwad College of Engineering

Assignment-05

Roll No: 123M1H010

Name of Student: Harshal Bhamare

Submission Date: 26 / 10 / 2024

1. Develop an Android application that allows the user to send and receive SMS messages. The app should have an input field for the phone number and message content. Upon clicking the Send button, the message should be sent to the specified phone number using the SMS Manager API. Additionally, implement a broadcast receiver to listen for incoming SMS messages and display the message content in a TextView.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

Activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/phone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Phone Number" />

    <EditText
        android:id="@+id/message"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Message" />

    <Button
        android:id="@+id/send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send" />

    <TextView
        android:id="@+id/received"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="" />
</LinearLayout>
```

MainActivity.java

```
package com.example.forpractice;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
import android.core.app.ActivityCompat;
import android.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity {

    private static final int MY_PERMISSIONS_REQUEST_SEND_SMS = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        EditText phoneT = findViewById(R.id.phone);
        EditText messageT = findViewById(R.id.message);
        Button send = findViewById(R.id.send);
        TextView received = findViewById(R.id.received);

        send.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String phone = phoneT.getText().toString();
                String message = messageT.getText().toString();

                if (ContextCompat.checkSelfPermission(MainActivity.this,
                        Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this,
                            new String[]{Manifest.permission.SEND_SMS},
                            MY_PERMISSIONS_REQUEST_SEND_SMS);
                } else {
                    SmsManager smsManager = SmsManager.getDefault();
                    smsManager.sendTextMessage(phone, null, message, null,
                            null);
                }
            }
        });
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        switch (requestCode) {
            case MY_PERMISSIONS_REQUEST_SEND_SMS: {
                if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {

```

```
        } else {
            }
        }
    }
}
```

SmsReceiver.java

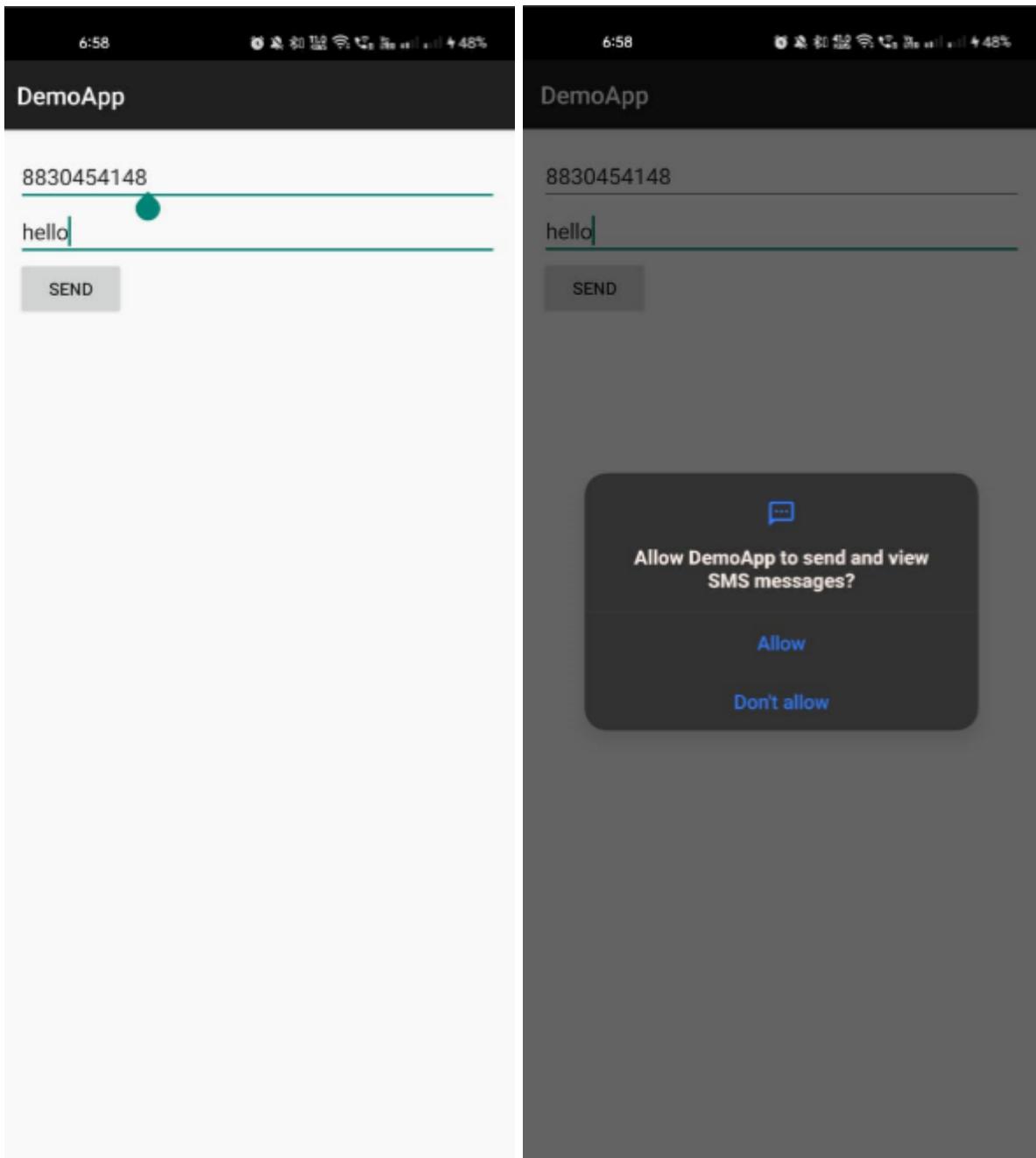
```
package com.example.forpractice;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.widget.TextView;

public class SmsReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        Bundle bundle = intent.getExtras();
        SmsMessage[] msgs = null;
        String str = "";

        if (bundle != null) {
            Object[] pdus = (Object[]) bundle.get("pdus");
            msgs = new SmsMessage[pdus.length];
            for (int i = 0; i < msgs.length; i++) {
                msgs[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
                str += "SMS from " + msgs[i].getOriginatingAddress();
                str += ":" + msgs[i].getMessageBody().toString();
                str += "\n";
            }
            TextView receivedMessage = ((MainActivity)
context).findViewById(R.id.received);
            receivedMessage.setText(str);
        }
    }
}
```



1. Create an application that sends an SMS message. When the user clicks the "SEND" button, a permission dialog should appear asking for permission to send and view SMS messages. If the user grants permission, the message should be sent successfully. If denied, the application should handle the denial gracefully.
2. Create an application that opens the camera interface to capture photos. Once the photo is taken, it should be displayed in an ImageView on the apps main screen. Use the Camera API or Intent with ACTION_IMAGE_CAPTURE to invoke the device's camera. Ensure proper handling of the permissions required for accessing the camera.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```

        android:orientation="vertical"
        android:padding="16dp">

    <Button
        android:id="@+id	btnCapture"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Capture Photo" />

    <ImageView
        android:id="@+id/imgPhoto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:scaleType="centerCrop" />
</LinearLayout>

```

MainActivity.xml

```

package com.example.forpractice;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_IMAGE_CAPTURE = 1;
    private ImageView imgPhoto;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

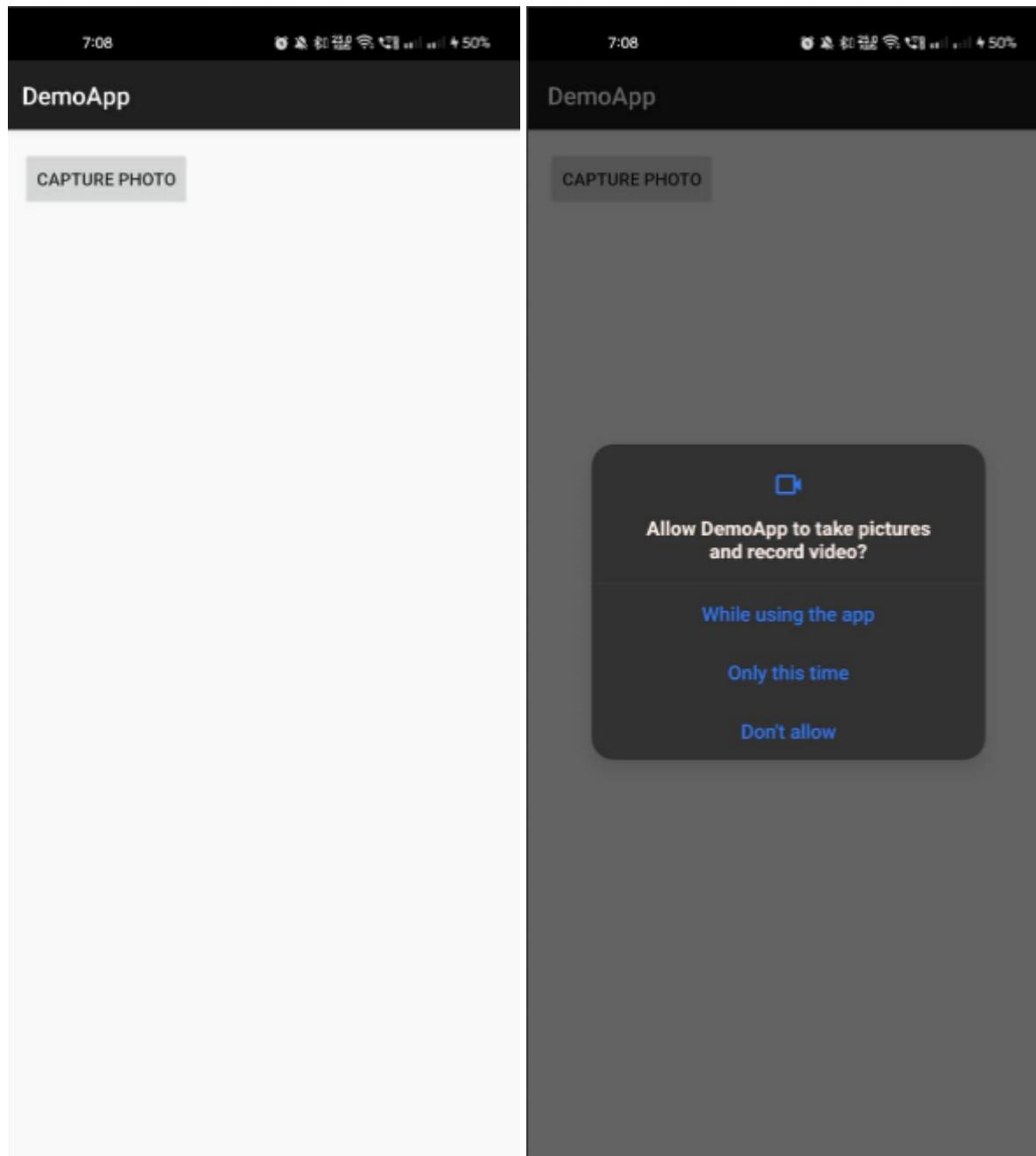
        Button btnCapture = findViewById(R.id.btnCapture);
        imgPhoto = findViewById(R.id.imgPhoto);

        btnCapture.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.CAMERA}, REQUEST_IMAGE_CAPTURE);
                } else {
                    dispatchTakePictureIntent();
                }
            }
        });
    }
}

```

```
private void dispatchTakePictureIntent() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        Bundle extras = data.getExtras();
        Bitmap imageBitmap = (Bitmap) extras.get("data");
        imgPhoto.setImageBitmap(imageBitmap);
    }
}
```



3. Design an Android app that allows users to initiate phone calls by entering a phone number and clicking a “Call” button. Additionally, implement functionality to listen for changes in call states (e.g., ringing, answered, idle) using the Telephony Manager API. Display the current call state in a TextView when it changes.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

Activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/etNum"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Phone Number" />

    <Button
        android:id="@+id	btnCall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Call" />

    <TextView
        android:id="@+id/tvState"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Call State" />
</LinearLayout>
```

MainActivity.java

```
package com.example.forpractice;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.telephony.TelephonyCallback;
import android.telephony.TelephonyManager;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity {
```

```

private static final int REQUEST_CALL_PERMISSION = 1;
private static final int REQUEST_PHONE_STATE_PERMISSION = 2;
private EditText etNum;
private TextView tvState;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    etNum = findViewById(R.id.etNum);
    Button btnCall = findViewById(R.id.btnCall);
    tvState = findViewById(R.id.tvState);

    btnCall.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(MainActivity.this, new
String[] {Manifest.permission.CALL_PHONE}, REQUEST_CALL_PERMISSION);
            } else {
                makeCall();
            }
        }
    });

    if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.READ_PHONE_STATE) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(MainActivity.this, new
String[] {Manifest.permission.READ_PHONE_STATE},
REQUEST_PHONE_STATE_PERMISSION);
    } else if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
        listenForCallStateChanges();
    }
}

private void makeCall() {
    String phoneNumber = etNum.getText().toString();
    Intent callIntent = new Intent(Intent.ACTION_CALL);
    callIntent.setData(Uri.parse("tel:" + phoneNumber));
    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) == PackageManager.PERMISSION_GRANTED) {
        startActivity(callIntent);
    }
}

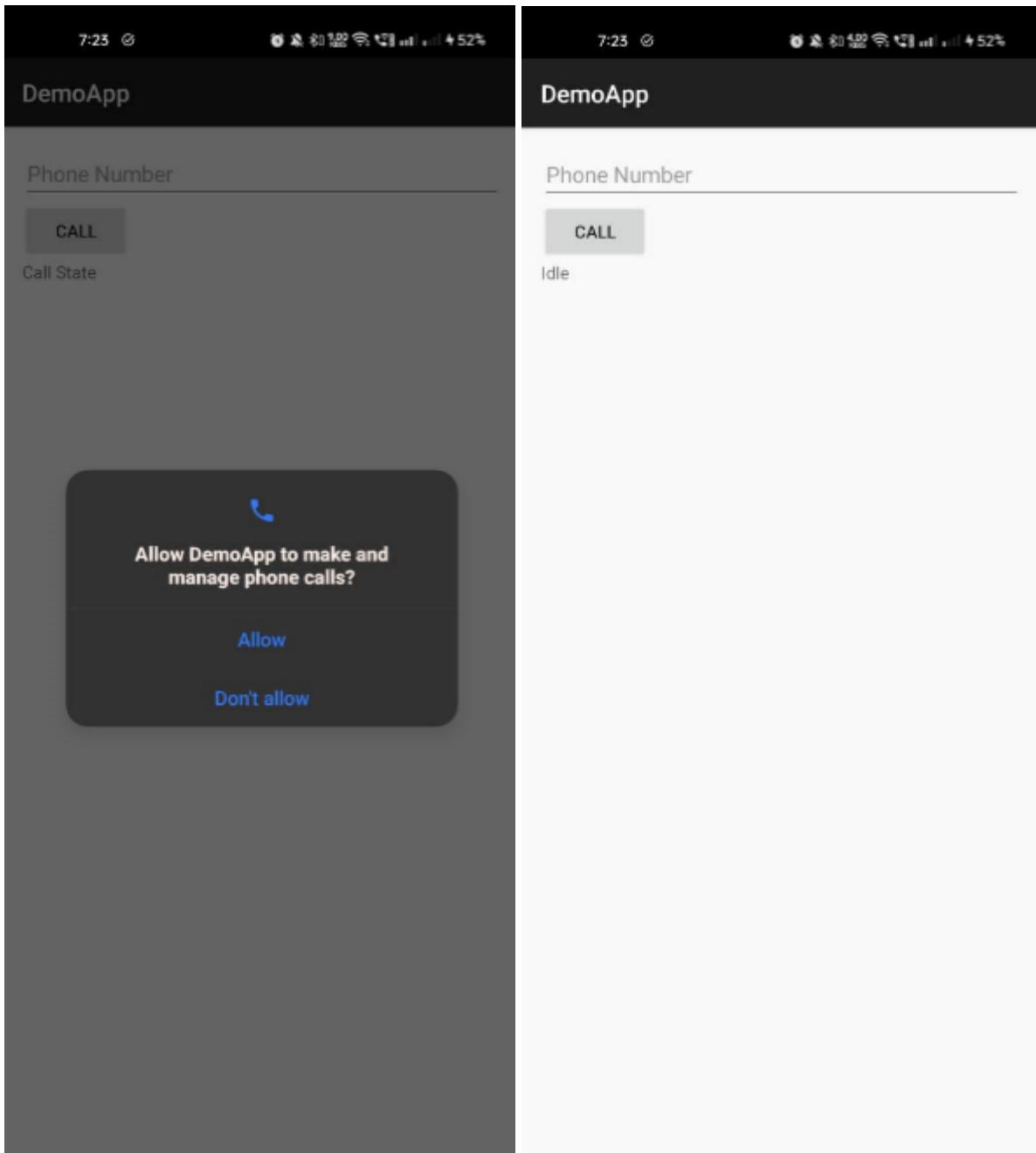
@RequiresApi(api = Build.VERSION_CODES.S)
private void listenForCallStateChanges() {
    TelephonyManager telephonyManager = (TelephonyManager)
getSystemService(TELEPHONY_SERVICE);
    if (telephonyManager != null) {
        telephonyManager.registerTelephonyCallback(getMainExecutor(), new
CustomTelephonyCallback());
    } else {
        Log.e("MainActivity", "TelephonyManager is null");
    }
}

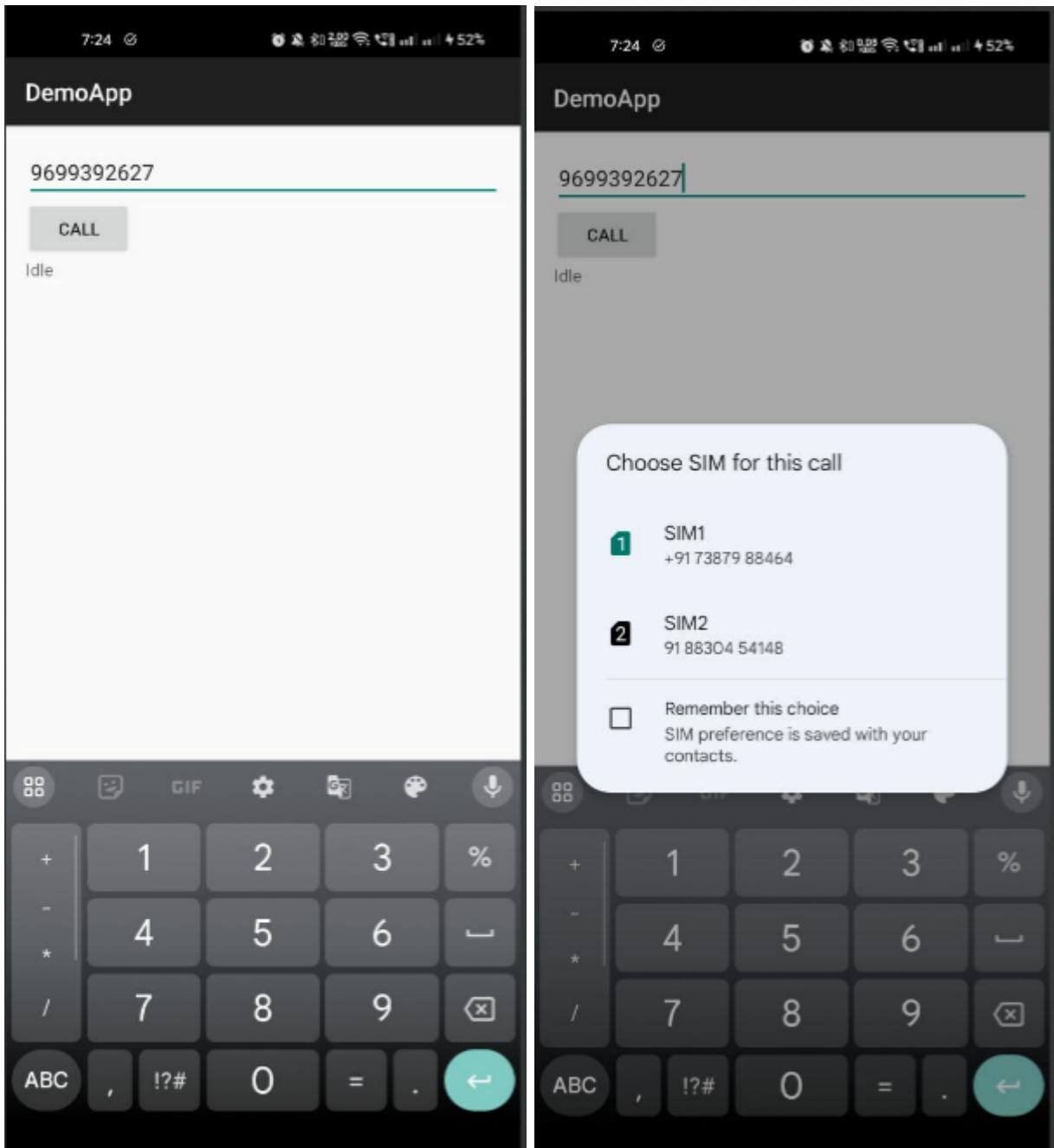
@RequiresApi(api = Build.VERSION_CODES.S)
private class CustomTelephonyCallback extends TelephonyCallback implements
TelephonyCallback.CallStateListener {

```

```
@Override
public void onCallStateChanged(int state) {
    switch (state) {
        case TelephonyManager.CALL_STATE_RINGING:
            tvState.setText("Ringing");
            break;
        case TelephonyManager.CALL_STATE_OFFHOOK:
            tvState.setText("Answered");
            break;
        case TelephonyManager.CALL_STATE_IDLE:
            tvState.setText("Idle");
            break;
        default:
            Log.e("CustomTelephonyCallback", "Unknown call state: " +
state);
            break;
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == REQUEST_PHONE_STATE_PERMISSION) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
                listenForCallStateChanges();
            }
        } else {
            Log.e("MainActivity", "Phone state permission denied");
        }
    } else if (requestCode == REQUEST_CALL_PERMISSION) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            makeCall();
        } else {
            Log.e("MainActivity", "Call permission denied");
        }
    }
}
```





4. Create a voice command application that uses the Speech API to recognize spoken words and convert them into text. The app should have a “Start Listening” button that initiates speech recognition, and the recognized text should be displayed in a TextView. Provide functionality for handling errors or when speech input is not detected.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

Activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    android:padding="16dp">
```

```

<Button
    android:id="@+id/listen"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Start Listening" />

<TextView
    android:id="@+id/result"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="" />
</LinearLayout>

```

MainActivity.java

```

package com.example.forpractice;

import android.Manifest;
import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.speech.SpeechRecognizer;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import java.util.ArrayList;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_RECORD_AUDIO_PERMISSION = 1;
    private static final int REQ_CODE_SPEECH_INPUT = 2;
    private TextView result;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        result = findViewById(R.id.result);
        Button btnListen = findViewById(R.id.listen);

        btnListen.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ContextCompat.checkSelfPermission(MainActivity.this,
                        Manifest.permission.RECORD_AUDIO) != PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this, new
                            String[]{Manifest.permission.RECORD_AUDIO}, REQUEST_RECORD_AUDIO_PERMISSION);
                } else {
                    startListening();
                }
            }
        });
    }

    private void startListening() {
        Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
                RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        intent.putExtra(RecognizerIntent.EXTRA_CALLING_PACKAGE,
                "com.example.forpractice");
        intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Say something");
        startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQ_CODE_SPEECH_INPUT) {
            if (resultCode == RESULT_OK) {
                ArrayList<String> resultText = data.getStringArrayListExtra(
                        RecognizerIntent.EXTRA_RESULTS);
                result.setText(resultText.get(0));
            }
        }
    }
}

```

```
        }
    }

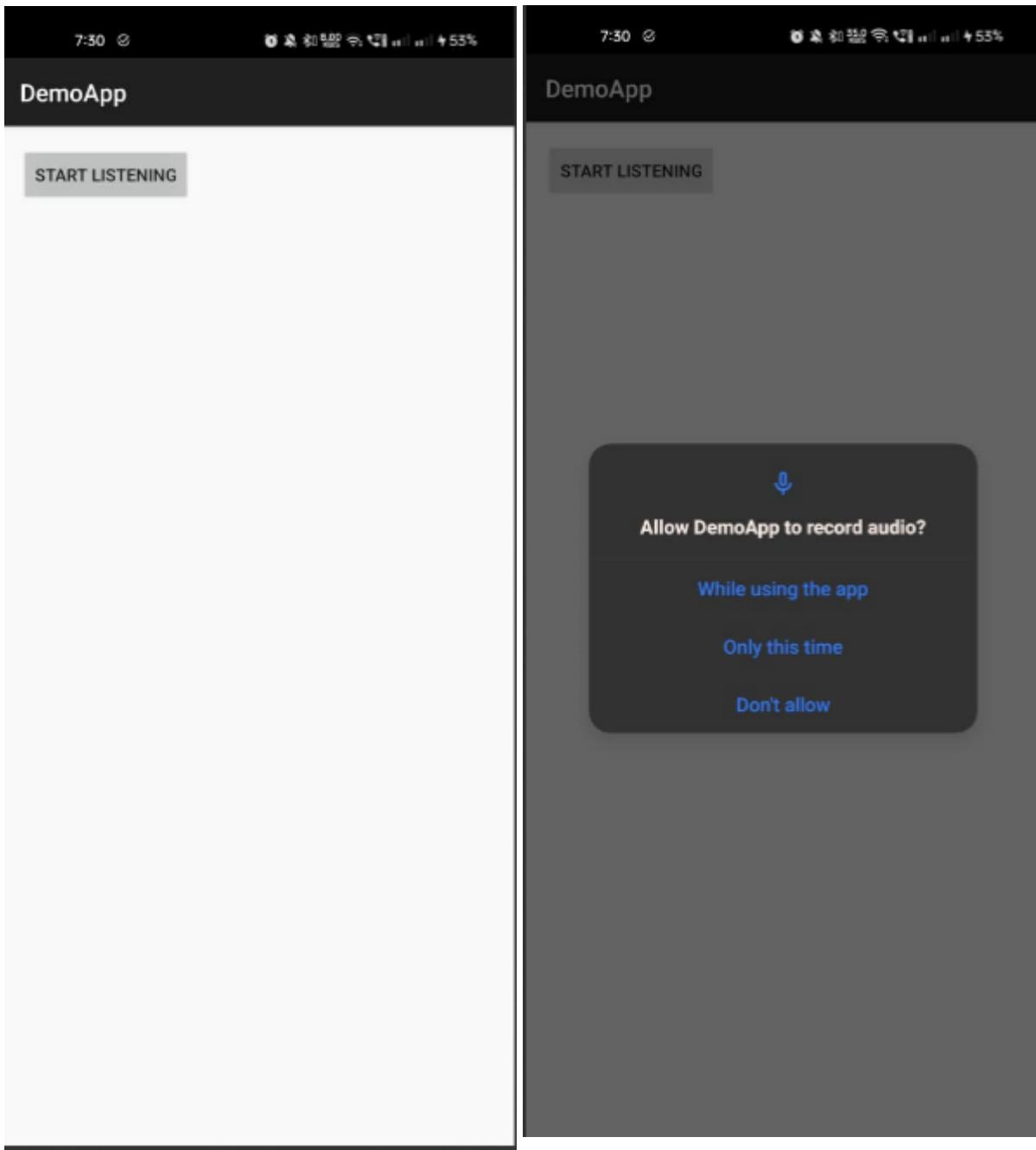
    private void startListening() {
        Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
        intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Speak now...");

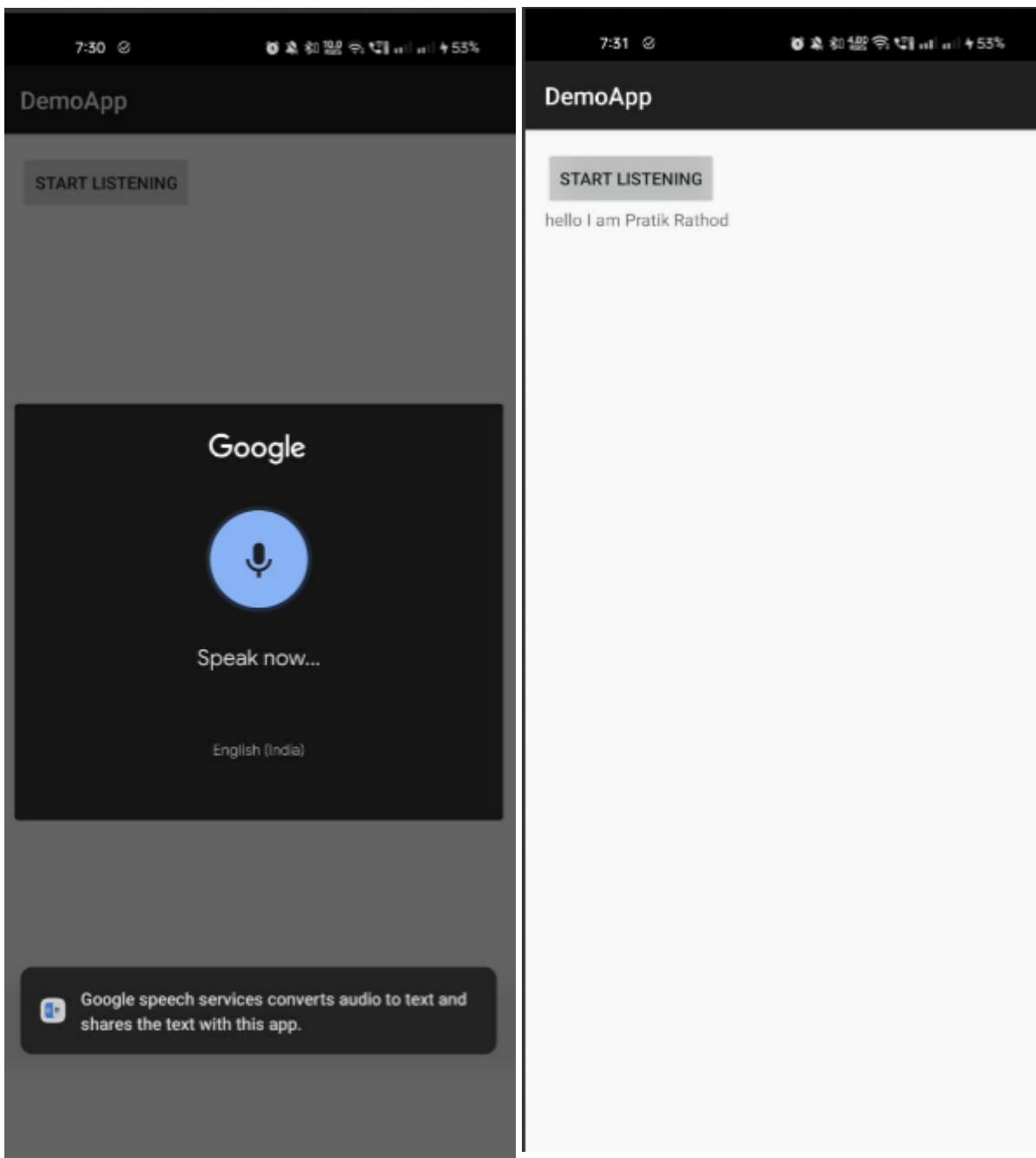
        try {
            startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
        } catch (ActivityNotFoundException a) {
            Toast.makeText(getApplicationContext(), "Speech recognition not
supported", Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);

        switch (requestCode) {
            case REQ_CODE_SPEECH_INPUT: {
                if (resultCode == RESULT_OK && data != null) {
                    ArrayList<String> res =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    if (res != null && !res.isEmpty()) {
                        result.setText(res.get(0));
                    } else {
                        result.setText("No speech detected");
                    }
                } else {
                    result.setText("Recognition error");
                }
                break;
            }
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == REQUEST_RECORD_AUDIO_PERMISSION) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                startListening();
            } else {
                Toast.makeText(this, "Permission denied",
Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```





5. Develop an application that retrieves and displays the user's current location (latitude and longitude) using the Location API. Use either FusedLocationProviderClient or LocationManager to obtain the location data. Display the location in a TextView and provide a button that refreshes the location. Additionally, show the location on a map using Google Maps API.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/refresh"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Refresh Location" />

    <TextView
        android:id="@+id/res"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Location: " />

    <fragment
        android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:name="com.google.android.gms.maps.SupportMapFragment" />
</LinearLayout>
```

MainActivity.xml

```

package com.example.forpractice;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class MainActivity extends AppCompatActivity implements
OnMapReadyCallback {

    private static final int REQUEST_LOCATION_PERMISSION = 1;
    private FusedLocationProviderClient fusedLocationClient;
    private TextView res;
    private GoogleMap map;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("MainActivity", "onCreate started");

        res = findViewById(R.id.res);
        Button refresh = findViewById(R.id.refresh);

        fusedLocationClient =
LocationServices.getFusedLocationProviderClient(this);
        Log.d("MainActivity", "FusedLocationProviderClient initialized");
```

```

refresh.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Log.d("MainActivity", "Refresh button clicked");
        if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED)
{
            ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION},
REQUEST_LOCATION_PERMISSION);
        } else {
            getLocation();
        }
    }
});

SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map);
if (mapFragment != null) {
    mapFragment.getMapAsync(this);
} else {
    Log.e("MainActivity", "MapFragment is null");
}

Log.d("MainActivity", "onCreate finished");
}

private void getLocation() {
try {
    Log.d("MainActivity", "Attempting to get location");
    fusedLocationClient.getLastLocation()
        .addOnSuccessListener(this, location -> {
            if (location != null) {
                double lat = location.getLatitude();
                double lng = location.getLongitude();
                res.setText("Location: " + lat + ", " + lng);
                LatLng latLng = new LatLng(lat, lng);
                map.addMarker(new
MarkerOptions().position(latLng).title("You are here"));
                map.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng, 15f));
            } else {
                res.setText("Unable to retrieve location");
                Log.d("MainActivity", "Location is null");
            }
        });
} catch (SecurityException e) {
    Log.e("MainActivity", "SecurityException: " + e.getMessage());
}
}

@Override
public void onMapReady(GoogleMap googleMap) {
    map = googleMap;
    Log.d("MainActivity", "Map is ready");
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == REQUEST_LOCATION_PERMISSION) {

```

```

        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            getLocation();
        } else {
            Toast.makeText(this, "Permission denied",
Toast.LENGTH_SHORT).show();
            Log.d("MainActivity", "Permission denied");
        }
    }
}

```

6. Build an application that sends SMS messages to a specified phone number. Ensure the app properly requests and handles SMS permissions at runtime. Implement functionality to show a confirmation message or status update in a TextView after sending the SMS. Also, handle scenarios where the user denies the permission and provide an appropriate message to the user.

AndroidManifest.xml

```

<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>

```

Activity_main.xml

```

<?xml version="1.0" encoding="utf-8" ?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/num"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Phone Number" />

    <EditText
        android:id="@+id/msg"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Message" />

    <Button
        android:id="@+id/send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send" />

    <TextView
        android:id="@+id/status"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Status" />
</LinearLayout>

```

MainActivity.xml

```
package com.example.forpractice;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_SEND_SMS = 1;
    private EditText num;
    private EditText msg;
    private TextView status;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        num = findViewById(R.id.num);
        msg = findViewById(R.id.msg);
        status = findViewById(R.id.status);
        Button send = findViewById(R.id.send);

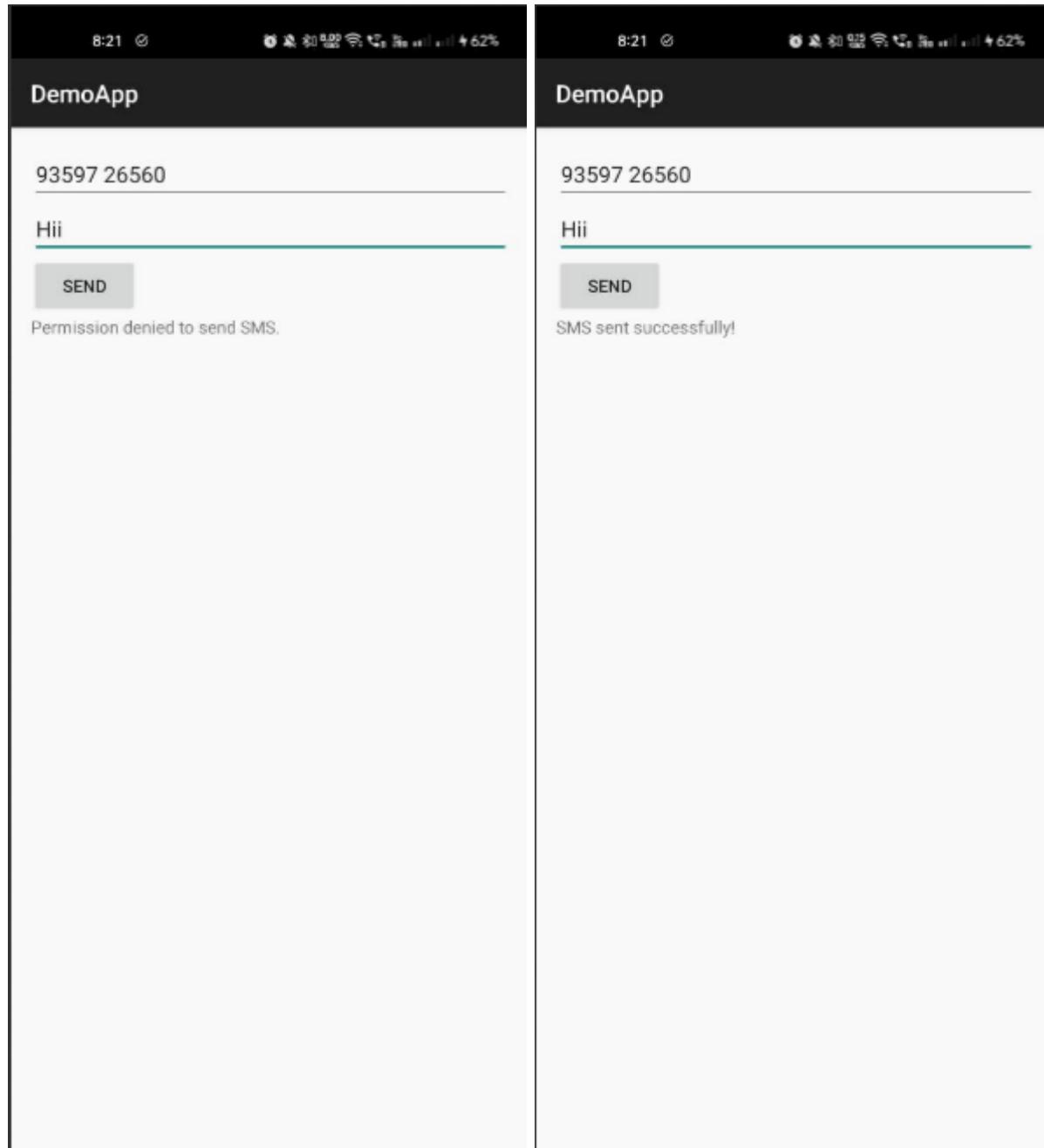
        send.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ContextCompat.checkSelfPermission(MainActivity.this,
                        Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this, new
                            String[]{Manifest.permission.SEND_SMS}, REQUEST_SEND_SMS);
                } else {
                    sendSMS();
                }
            }
        });
    }

    private void sendSMS() {
        String phoneNumber = num.getText().toString();
        String message = msg.getText().toString();

        try {
            SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(phoneNumber, null, message, null, null);
            status.setText("SMS sent successfully!");
        } catch (Exception e) {
            status.setText("Failed to send SMS.");
            e.printStackTrace();
        }
    }

    @Override
```

```
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == REQUEST_SEND_SMS) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            sendSMS();
        } else {
            status.setText("Permission denied to send SMS.");
        }
    }
}
```



7. Design an app that captures a photo using the device's camera and saves it to the external storage. After taking the photo, the app should display it in an ImageView and save the photo

to a specified directory. Implement proper handling of storage permissions and ensure the photo is stored with a unique filename to avoid overwriting existing files.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/capture"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Capture Photo" />

    <ImageView
        android:id="@+id/photo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:scaleType="centerCrop" />
</LinearLayout>
```

MainActivity.java

```
package com.example.forpractice;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.core.content.FileProvider;
import java.io.File;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {
```

```
private static final int REQUEST_IMAGE_CAPTURE = 1;
private static final int REQUEST_PERMISSIONS = 2;
private ImageView photo;
private String currentPhotoPath;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    photo = findViewById(R.id.photo);
    Button capture = findViewById(R.id.capture);

    capture.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED ||
                ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) != PackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(MainActivity.this, new
String[] {Manifest.permission.CAMERA,
Manifest.permission.WRITE_EXTERNAL_STORAGE}, REQUEST_PERMISSIONS);
            } else {
                dispatchTakePictureIntent();
            }
        }
    });
}

private void dispatchTakePictureIntent() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        File photoFile = null;
        try {
            photoFile = createImageFile();
        } catch (IOException ex) {
            Log.e("MainActivity", "Error occurred while creating the file: " + ex.getMessage());
        }
        if (photoFile != null) {
            Uri photoURI = FileProvider.getUriForFile(this,
"com.example.android.fileprovider", photoFile);
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
            startActivityForResult(takePictureIntent,
REQUEST_IMAGE_CAPTURE);
        }
    }
}

private File createImageFile() throws IOException {
    String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss",
Locale.getDefault()).format(new Date());
    String imageFileName = "JPEG_" + timeStamp + "_";
    File storageDir = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
    File image = File.createTempFile(imageFileName, ".jpg", storageDir);
    currentPhotoPath = image.getAbsolutePath();
    return image;
}

@Override
```

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        setPic();
    }
}

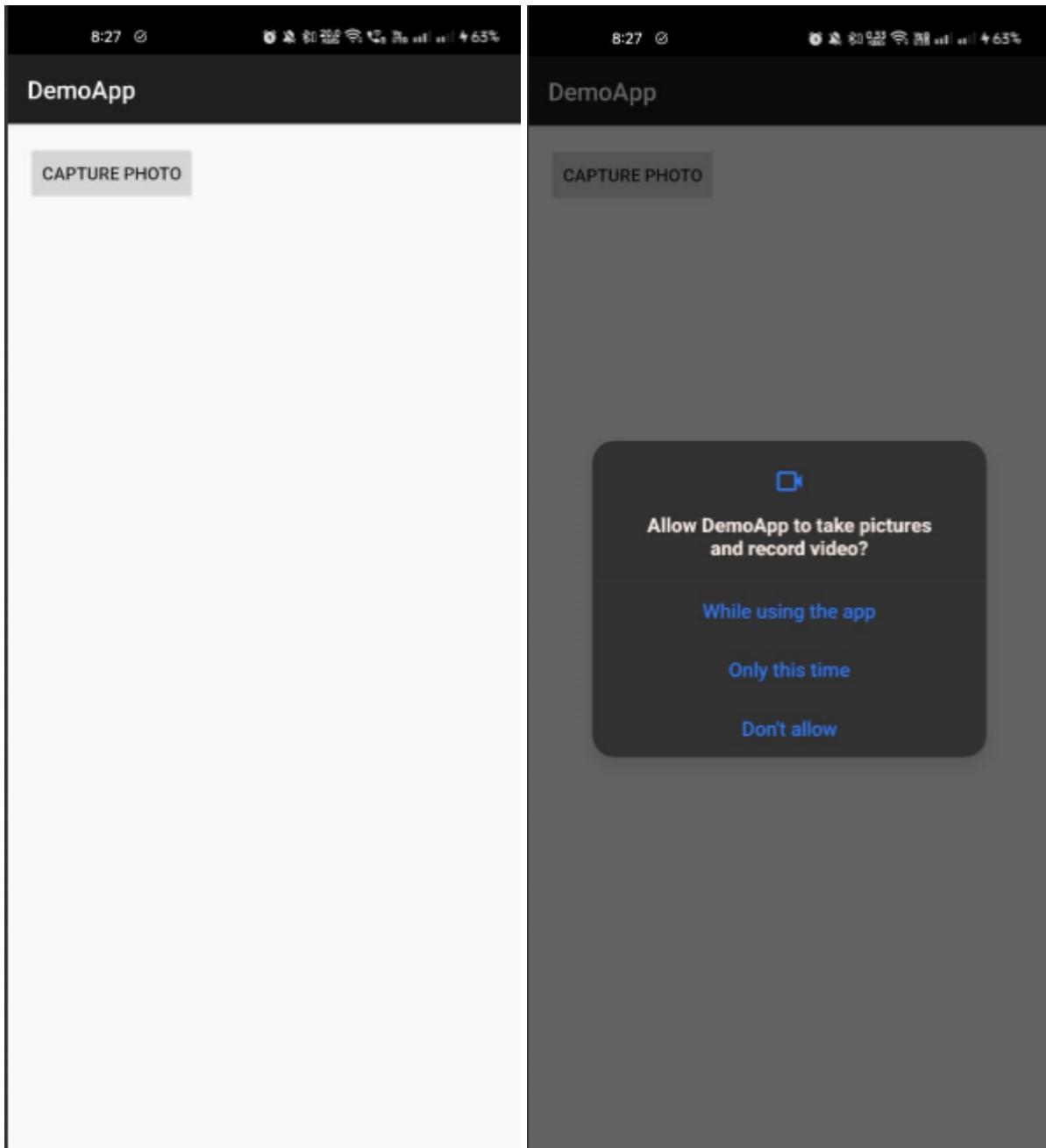
private void setPic() {
    int targetW = photo.getWidth();
    int targetH = photo.getHeight();

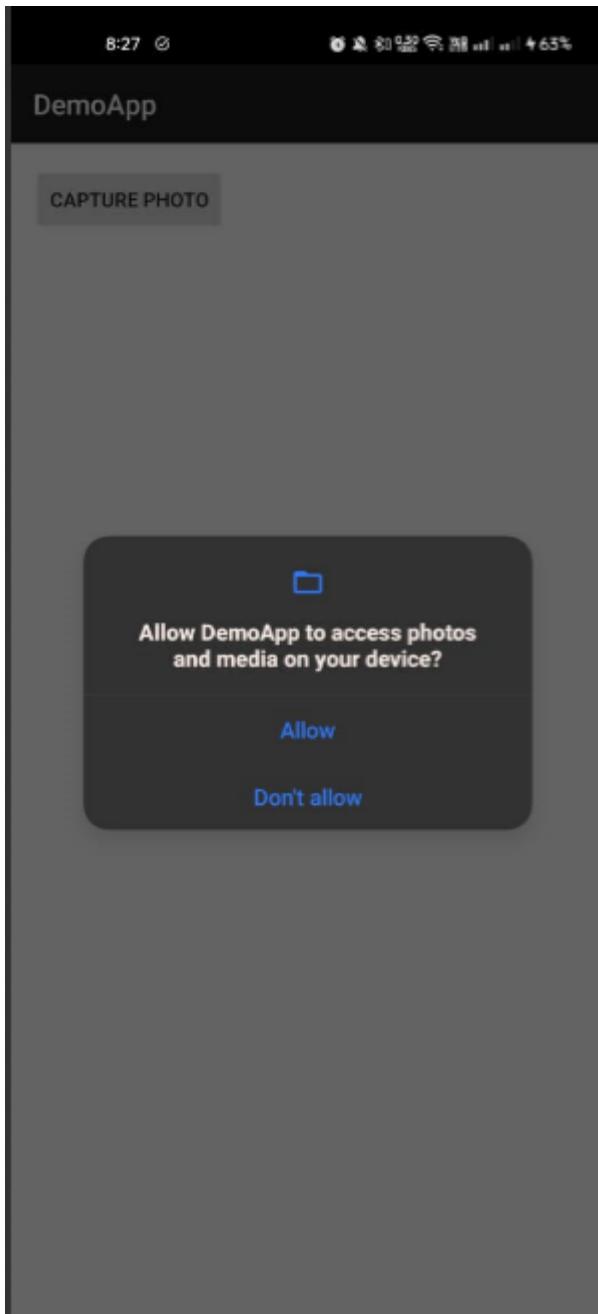
    BitmapFactory.Options bmOptions = new BitmapFactory.Options();
    bmOptions.inJustDecodeBounds = true;
    BitmapFactory.decodeFile(currentPhotoPath, bmOptions);
    int photoW = bmOptions.outWidth;
    int photoH = bmOptions.outHeight;

    int scaleFactor = Math.min(photoW / targetW, photoH / targetH);

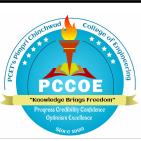
    bmOptions.inJustDecodeBounds = false;
    bmOptions.inSampleSize = scaleFactor;
    Bitmap bitmap = BitmapFactory.decodeFile(currentPhotoPath, bmOptions);
    photo.setImageBitmap(bitmap);
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == REQUEST_PERMISSIONS) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            dispatchTakePictureIntent();
        } else {
            Toast.makeText(this, "Permission denied",
Toast.LENGTH_SHORT).show();
        }
    }
}
```





8. Create an application that monitors both incoming and outgoing phone calls. Use the Telephony API to listen for call state changes and record details such as the callers phone number and call duration. Display this information in a ListView or RecyclerView, and ensure the app handles call logs and permissions appropriately.
9. Develop an app that uses speech recognition to convert spoken words into text and provides spoken feedback using Text-to-Speech. Implement a button to start speech recognition and another button to convert text into speech. Display the recognized text in a TextView and use Text-to Speech to read the text aloud when the user clicks the corresponding button.
10. Create an application that tracks the users location and calculates the distance traveled between two points. Use the Location API to obtain the user's current location at different intervals. Implement functionality to calculate the distance between the starting location and the current location and display this distance in a TextView.



Assignment-06

Roll No: 123M1H010

Name of Student: Harshal Bhamare

Submission Date: 30 / 10/ 2024

1. Create an Android application that demonstrates file management in internal storage. Implement functionality to save a text file containing user input to internal storage when a button is clicked. Provide options to read from and delete the saved file. Ensure that the file operations handle cases where the file does not exist and display appropriate messages to the user

Solution:

XML FILE:

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="write"
    android:id="@+id/write"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="delete"
    android:id="@+id/delete"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="read"
    android:id="@+id/read"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/fdata"/>
</LinearLayout>
    android:layout_width="wrap_content"
    android:hint="enter data:"/>
```

JAVA FILE:

```
package com.example.assign6;

import android.content.Context; import
android.content.pm.PackageManager;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.widget.*;
import android.view.*;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
```

```
public class MainActivity extends AppCompatActivity {

    Button write, delete, read;

    EditText fname, data;
    TextView fdata;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
        write = findViewById(R.id.write);
        read = findViewById(R.id.read);
        delete = findViewById(R.id.delete);
        fname = findViewById(R.id.fname);
        data = findViewById(R.id.data);
        fdata = findViewById(R.id.fdata);

        write.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String f = fname.getText().toString();
                String d = data.getText().toString();
```

```

        try{
            FileOutputStream fos = openFileOutput(f,
Context.MODE_PRIVATE);
            fos.write(d.getBytes());
            Toast.makeText(MainActivity.this, "Written",
Toast.LENGTH_SHORT).show();
        }
        catch(Exception e){
            e.printStackTrace();
        }
    }

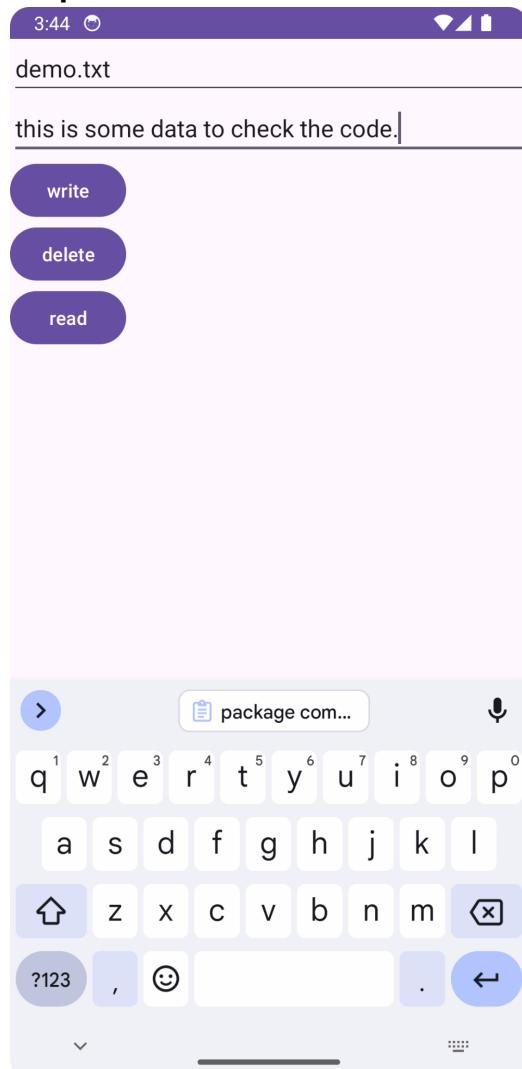
    read.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String f = fname.getText().toString();
            String l;
            try{
                FileInputStream fis = openFileInput(f);
                InputStreamReader isr = new InputStreamReader(fis);
                BufferedReader br = new BufferedReader(isr);
                StringBuilder sb = new StringBuilder();
                while((l = br.readLine())!=null){
                    sb.append(l);
                }
                fdata.setText(sb.toString());
                Toast.makeText(MainActivity.this, "Read done",
Toast.LENGTH_SHORT).show();
            }
            catch(Exception e){
                e.printStackTrace();
            }
        }
    });

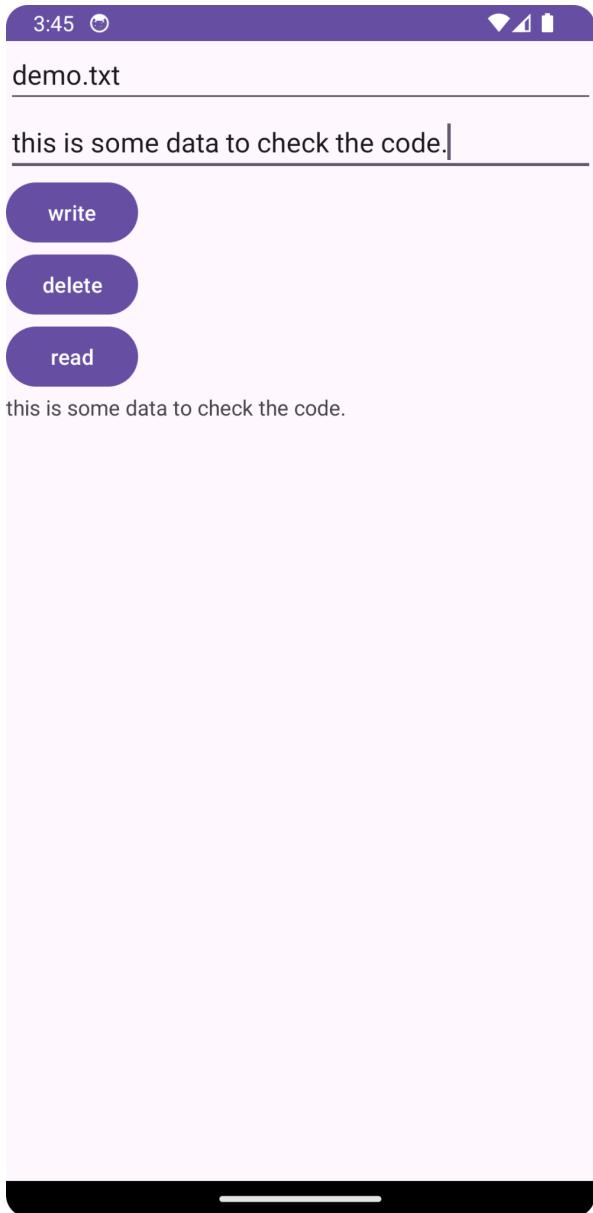
    delete.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String f = fname.getText().toString();
            if(deleteFile(f)){
                Toast.makeText(MainActivity.this, "Delete",
Toast.LENGTH_SHORT).show();
            }
        }
    });
}

```

```
}
```

Output:



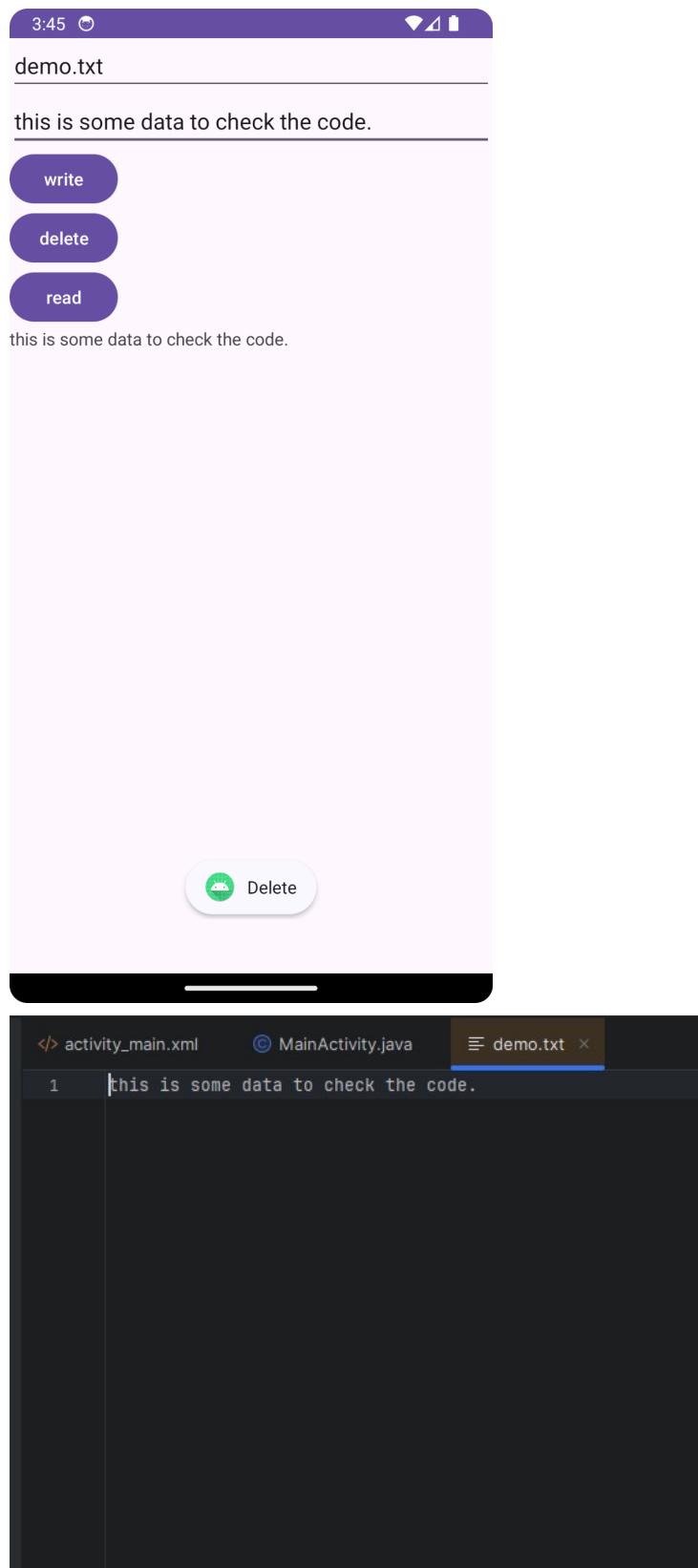


this is some data to check the code.

write

delete

read



2. Develop an app that allows users to save and retrieve files from external storage (e.g., SD card). Implement functionality to create a directory in external storage, save a text file with user input, and list all files in the directory. Ensure that the app properly requests and handles external storage permissions and provides feedback if the permissions are not granted.

Solution:

JAVA FILE:

```
package com.example.assign6;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.Settings;
import android.util.Log;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    private static final int PERMISSION_REQUEST_CODE = 100;
    private EditText inputText;
    private TextView fileList;
    private File directory;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        inputText = findViewById(R.id.inputText);
```

```
Button saveButton = findViewById(R.id.saveButton);  
  
Button listFilesButton = findViewById(R.id.listFilesButton);  
fileList = findViewById(R.id.fileList);  
  
// Request permissions if not granted  
if (!checkPermissions()) {  
    requestPermissions();  
} else {  
    createDirectory();  
}  
  
saveButton.setOnClickListener(v -> {  
    if (checkPermissions()) {  
        String content = editText.getText().toString();  
        if (!content.isEmpty()) {  
            saveToFile(content);  
        } else {  
            Toast.makeText(this, "Please enter some text",  
Toast.LENGTH_SHORT).show();  
        }  
    } else {  
        Toast.makeText(this, "Storage permission not granted",  
Toast.LENGTH_SHORT).show();  
    }  
});  
  
listFilesButton.setOnClickListener(v -> {  
    if (checkPermissions()) {  
       .listFilesInDirectory();  
    } else {  
        Toast.makeText(this, "Storage permission not granted",  
Toast.LENGTH_SHORT).show();  
    }  
});  
  
private boolean checkPermissions() {  
    if (android.os.Build.VERSION.SDK_INT >=  
        android.os.Build.VERSION_CODES.R) {  
        return Environment.isExternalStorageManager();  
    } else {  
        int readPermission = ContextCompat.checkSelfPermission(  
            this, Manifest.permission.READ_EXTERNAL_STORAGE);  
        return readPermission == PackageManager.PERMISSION_GRANTED;  
    }  
}
```

```
    }

    private void requestPermissions() {
        if (Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.R) {
            Intent intent = new Intent(Settings.ACTION_MANAGE_APP_ALL_FILES_ACCESS_PERMISSION);
            intent.setData(Uri.parse("package:" + getPackageName()));
            startActivityForResult(intent, PERMISSION_REQUEST_CODE);
        } else {
            ActivityCompat.requestPermissions(
                this,
                new String[] {Manifest.permission.READ_EXTERNAL_STORAGE},

```

```
                PERMISSION_REQUEST_CODE);
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                createDirectory();
                Toast.makeText(this, "Permission granted", Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(this, "Permission denied", Toast.LENGTH_SHORT).show();
            }
        }
    }

    private void createDirectory() { directory = new File(getExternalFilesDir(null), "MyAppFiles");
        if (!directory.exists()) {
            if (directory.mkdirs()) {
                Log.i("MainActivity", "Directory created: " + directory.getAbsolutePath());
            } else {
                Log.e("MainActivity", "Failed to create directory.");
                Toast.makeText(this, "Failed to create directory",
                    Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

```

        }
    } else {
        Log.i("MainActivity", "Directory already exists: " +
directory.getAbsolutePath());
    }
}

private void saveToFile(String content) {
    if (directory == null) {
        Toast.makeText(this, "Directory not available",
Toast.LENGTH_SHORT).show();
        return;
    }

    File file = new File(directory, "UserInput_" +
System.currentTimeMillis() + ".txt"); try (FileOutputStream
fos = new FileOutputStream(file)) {
        fos.write(content.getBytes());
        Toast.makeText(this, "File saved: " + file.getName(),
Toast.LENGTH_SHORT).show();
    } catch (IOException e) {
        Log.e("MainActivity", "Failed to save file", e);
        Toast.makeText(this, "Failed to save file",
Toast.LENGTH_SHORT).show();
    }
}

private void listFilesInDirectory() {
    if (directory == null) {
        Toast.makeText(this, "Directory not available",
Toast.LENGTH_SHORT).show();
        return;
    }

    StringBuilder builder = new StringBuilder();
    File[] files = directory.listFiles();

    if (files != null && files.length > 0) {
        for (File file : files) {
            builder.append(file.getName()).append("\n");
        }
    } else {
        builder.append("No files found.");
    }

    fileList.setText(builder.toString());
}
}

```

XML FILE:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <EditText
        android:id="@+id/inputText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter text to save" />

    <Button
        android:id="@+id/saveButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Save to File"
        android:layout_below="@id/inputText"
        android:layout_marginTop="16dp" />

    <Button
        android:id="@+id/listFilesButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="List Files"
        android:layout_below="@id/saveButton"
        android:layout_marginTop="16dp" />

    <TextView
        android:id="@+id/fileList"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/listFilesButton"
        android:layout_marginTop="16dp" />
</RelativeLayout>
```

Output:

11:38 ☀



All files access



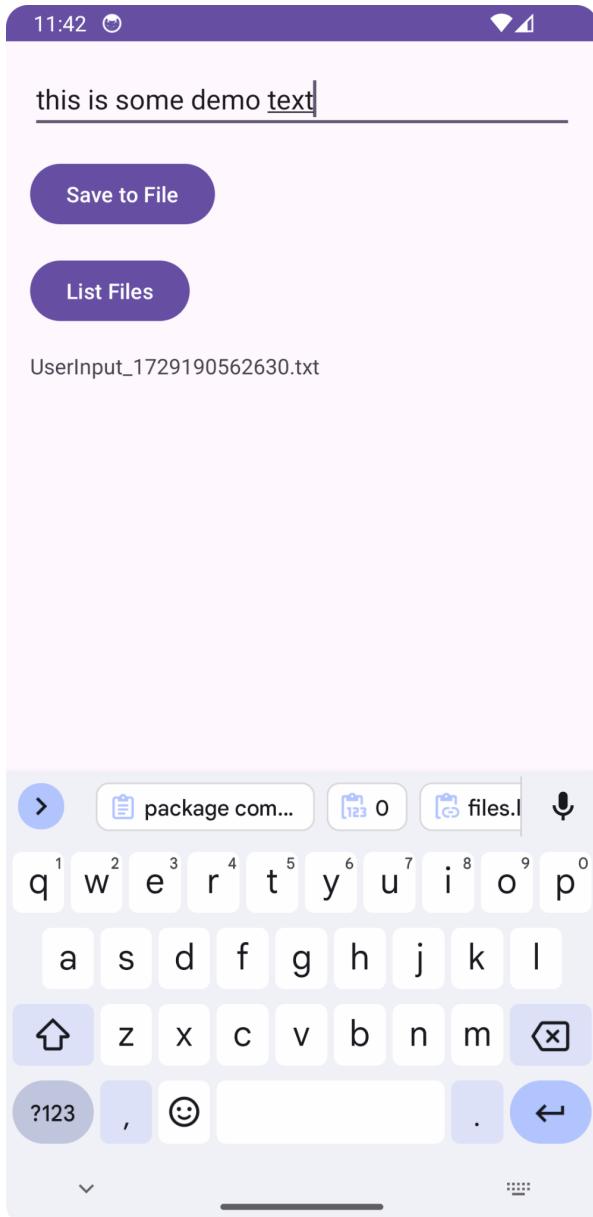
assign6

1.0

Allow access to manage all
files



Allow this app to read, modify and delete all files on this
device or any connected storage volumes. If granted,
app may access files without your explicit knowledge.



3. Build a simple notes application that uses SQLite to store and retrieve notes. Implement a database schema to store notes with fields for title and content. Create an activity that allows users to add, view, edit, and delete notes. Use SQLiteOpenHelper to manage database creation and version management, and provide a user-friendly interface for interacting with the notes.

Solution:

```
package com.example.assign6;
```

XML FILE:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:orientation="vertical">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/title"
        android:hint="enter title:"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/content"
        android:hint="enter content:"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="insert"
        android:id="@+id/insert"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="delete"
        android:id="@+id/delete"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="read"
        android:id="@+id/read"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="update"
        android:id="@+id/update"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/fdata"/>
</LinearLayout>
```

JAVA FILE:

```
import android.content.Context;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.widget.*;
import android.view.*;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity {

    Button insert, delete, read, update;

    EditText title, content;
    TextView fdata;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        insert = findViewById(R.id.insert);
        delete = findViewById(R.id.delete);
        read = findViewById(R.id.read);
        update = findViewById(R.id.update);
        title = findViewById(R.id.title);
        content = findViewById(R.id.content);
        fdata = findViewById(R.id.fdata);

        dbHelper db = new dbHelper(MainActivity.this);

        insert.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String t = title.getText().toString();
                String c = content.getText().toString();
                boolean f = db.insertData(t, c);
                if(f) {
                    Toast.makeText(MainActivity.this, "Inserted",
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

```

        }
    else{
        Toast.makeText(MainActivity.this, "Not Inserted",
Toast.LENGTH_SHORT).show();
    }
}
);

read.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Cursor res = db.getAllData();
        if (res != null && res.getCount() > 0) {
            StringBuilder stringBuilder = new StringBuilder();
            while (res.moveToNext()) {
                String title = res.getString(0);
                String content = res.getString(1);
                stringBuilder.append("Title:
").append(title).append("\n")
                    .append("Content:
");
                content).append("\n\n");
            }
            fdata.setText(stringBuilder.toString());
        }
    }
});

delete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String titleToDelete = title.getText().toString().trim();
        if (!titleToDelete.isEmpty()) { int rowsDeleted =
db.deleteData(titleToDelete);
        if (rowsDeleted > 0) {
            Toast.makeText(MainActivity.this, "Deleted",
Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(MainActivity.this, "No items
Deleted", Toast.LENGTH_SHORT).show();
        }
    }
}
);

update.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String titleToUpdate = title.getText().toString().trim();

```

```

        String newContent = content.getText().toString().trim();

        if (!titleToUpdate.isEmpty() && !newContent.isEmpty()) {
            boolean isUpdated = db.updateData(titleToUpdate,
newContent);
            if (isUpdated) {
                Toast.makeText(MainActivity.this, "Updated",
Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(MainActivity.this, "No updates",
Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(MainActivity.this, "Please enter both title
and new content", Toast.LENGTH_SHORT).show();
        }
    }
}
}

```

DATABASE HELPER:

```

package com.example.assign6;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class dbHelper extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = "Notes.db";
    public static final String TABLE_NAME = "NOTES";
    public static final String COL_1 = "TITLE ";
    public static final String COL_2 = "CONTENT";

    public dbHelper(Context context) {
        super(context, DATABASE_NAME, null, 1);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table " + TABLE_NAME +" (TITLE      TEXT,      CONTENT"
                + " TEXT)");
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
}
}

```

```

        db.execSQL(DROP TABLE IF EXISTS "+TABLE_NAME);
onCreate(db)

}

public boolean insertData(String t, String c) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(COL_1, t);
    contentValues.put(COL_2, c);

    try { long result = db.insert(TABLE_NAME, null,
        contentValues);
        return result != -1;
    } catch (Exception e) {
        Log.e("DB_ERROR", "Error inserting data: " + e.getMessage());
        return false;
    }
}

public Cursor getAllData() {
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor res = db.rawQuery("select * from "+TABLE_NAME,null);
    return res;
}

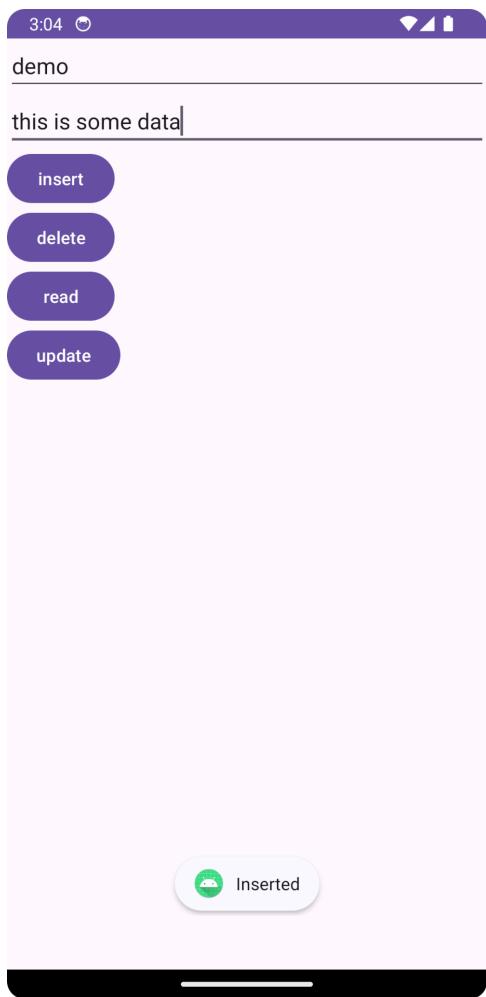
public boolean updateData(String t,String c) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(COL_1,t);
    contentValues.put(COL_2,c);
    db.update(TABLE_NAME, contentValues, "TITLE = ?",new String[] { t });
    return true;
}

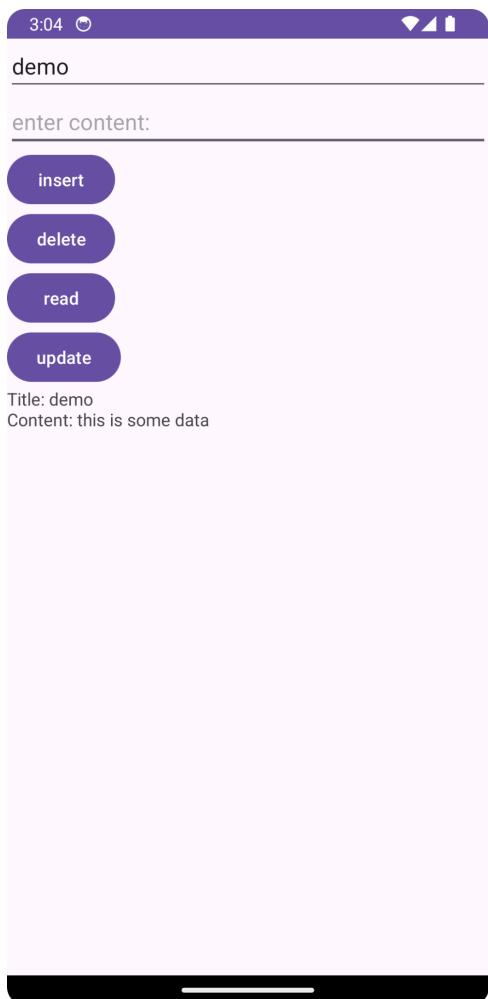
public Integer deleteData (String t) {
    SQLiteDatabase db = this.getWritableDatabase();
    return db.delete(TABLE_NAME, "TITLE = ?",new String[] {t}); }

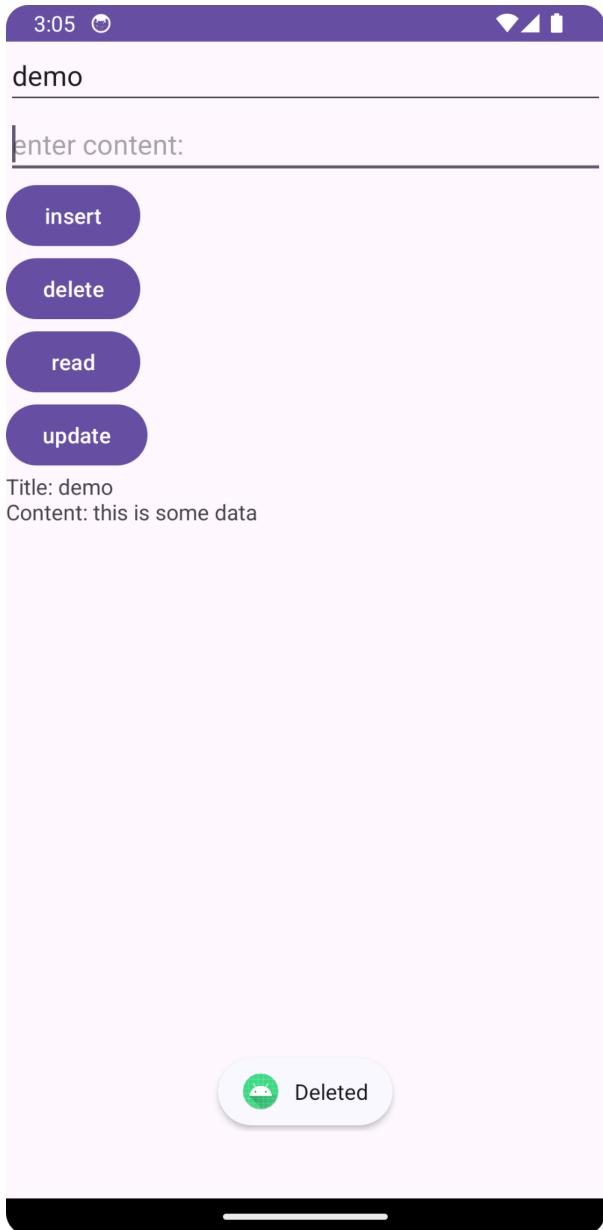
}

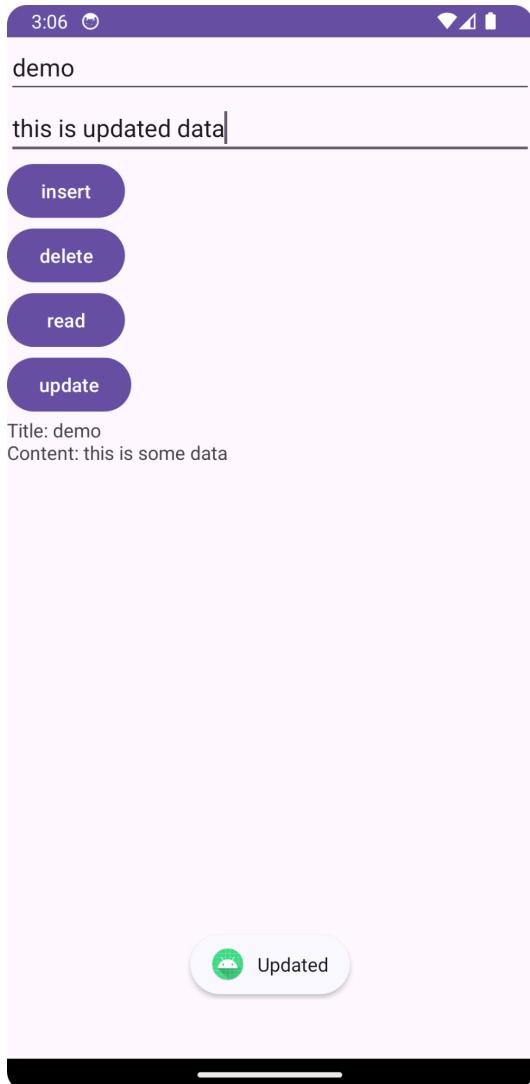
```

Output:









4. Design an application that uses Shared Preferences to save and retrieve user settings. Implement a settings screen where users can toggle options such as dark mode or notification preferences. Store these settings using Shared Preferences and apply them throughout the app. Provide functionality to reset preferences to default values and ensure that changes are reflected immediately in the app.

Solution:

XML FILE:

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
        android:id="@+id/main"
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Dark Mode:"/>

<ToggleButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn="ON"
    android:textOff="OFF"
    android:id="@+id/dark"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Notifications"/>

<ToggleButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn="ON"
    android:textOff="OFF"
    android:id="@+id/notific"/>

</LinearLayout>
```

JAVA FILE:

```
package com.example.myapplication;

import android.content.SharedPreferences;
import android.os.Bundle;
import android.widget.CompoundButton;
import android.widget.ToggleButton;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    ToggleButton dark, notific;
    SharedPreferences sp;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

dark = findViewById(R.id.dark);
notific = findViewById(R.id.notific);

sp = getSharedPreferences("prefs", MODE_PRIVATE);
SharedPreferences.Editor e = sp.edit(); if(sp.getInt("dark", 0)==1) {
    dark.setChecked(true);
}
if(sp.getInt("notific", 0)==1) {
    notific.setChecked(true);
}
dark.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
        if(isChecked) {
            e.putInt("dark", 1);
            e.apply();
        } else{
            e.putInt("dark", 0);
            e.apply(); }

    } });
}

notific.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
        if(isChecked) {
            e.putInt("notific", 1);
            e.apply(); }

        else{
            e.putInt("notific", 0);
            e.apply(); }
    }
});
```

```
}
```

Output:



5. Create an application that performs various file operations (create, read, update, delete) using internal storage. The app should allow users to create a file with some initial content, read the content and display it in a TextView, update the content with new data, and delete the file when no longer needed. Ensure that the app handles file operations gracefully and informs users of any errors.

Solution:

XML FILE:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:orientation="vertical">
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/fname"
    android:hint="enter file name:"/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/data"
    android:hint="enter data:"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="write"
    android:id="@+id/write"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="delete"
    android:id="@+id/delete"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="read"
    android:id="@+id/read"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/fdata"/>
</LinearLayout>
```

JAVA FILE:

```
package com.example.assign6;

import android.content.Context;
import android.content.pm.PackageManager;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.widget.*;
import android.view.*;
```

```
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity {
```

```
    Button write, delete, read;

    EditText fname, data;
    TextView fdata;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        write = findViewById(R.id.write);
        read = findViewById(R.id.read);
        delete = findViewById(R.id.delete);
        fname = findViewById(R.id.fname);
        data = findViewById(R.id.data);
        fdata = findViewById(R.id.fdata);

        write.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String f = fname.getText().toString();
                String d = data.getText().toString();

                try{
                    FileOutputStream fos = openFileOutput(f,
Context.MODE_PRIVATE);
                    fos.write(d.getBytes());
                    Toast.makeText(MainActivity.this, "Written",
Toast.LENGTH_SHORT).show();
                }
                catch(Exception e){
                    e.printStackTrace();
                }
            }
        });
    }
}
```

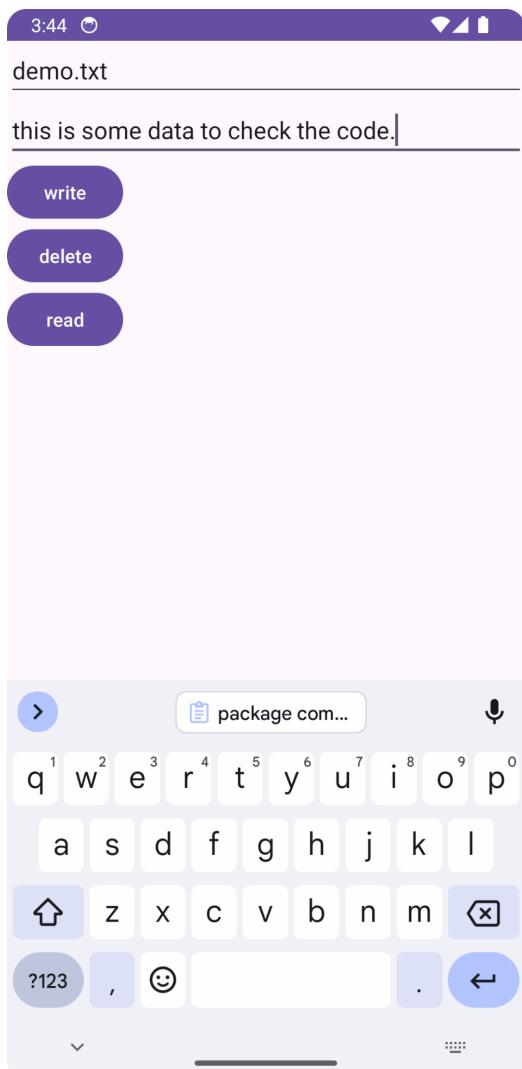
```
        }

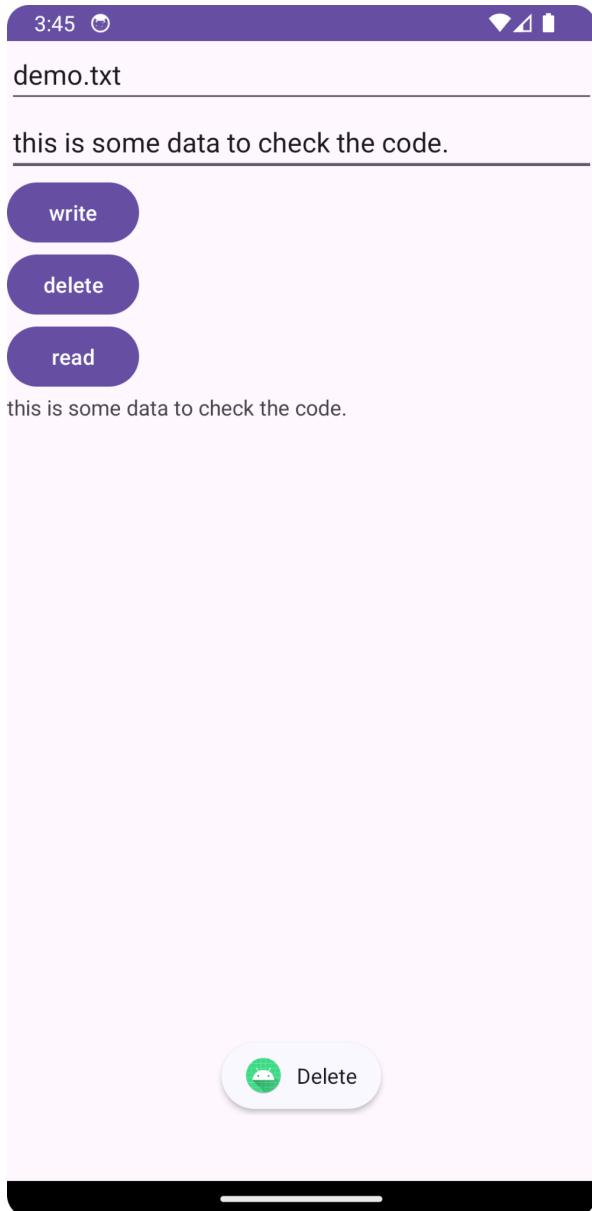
    }

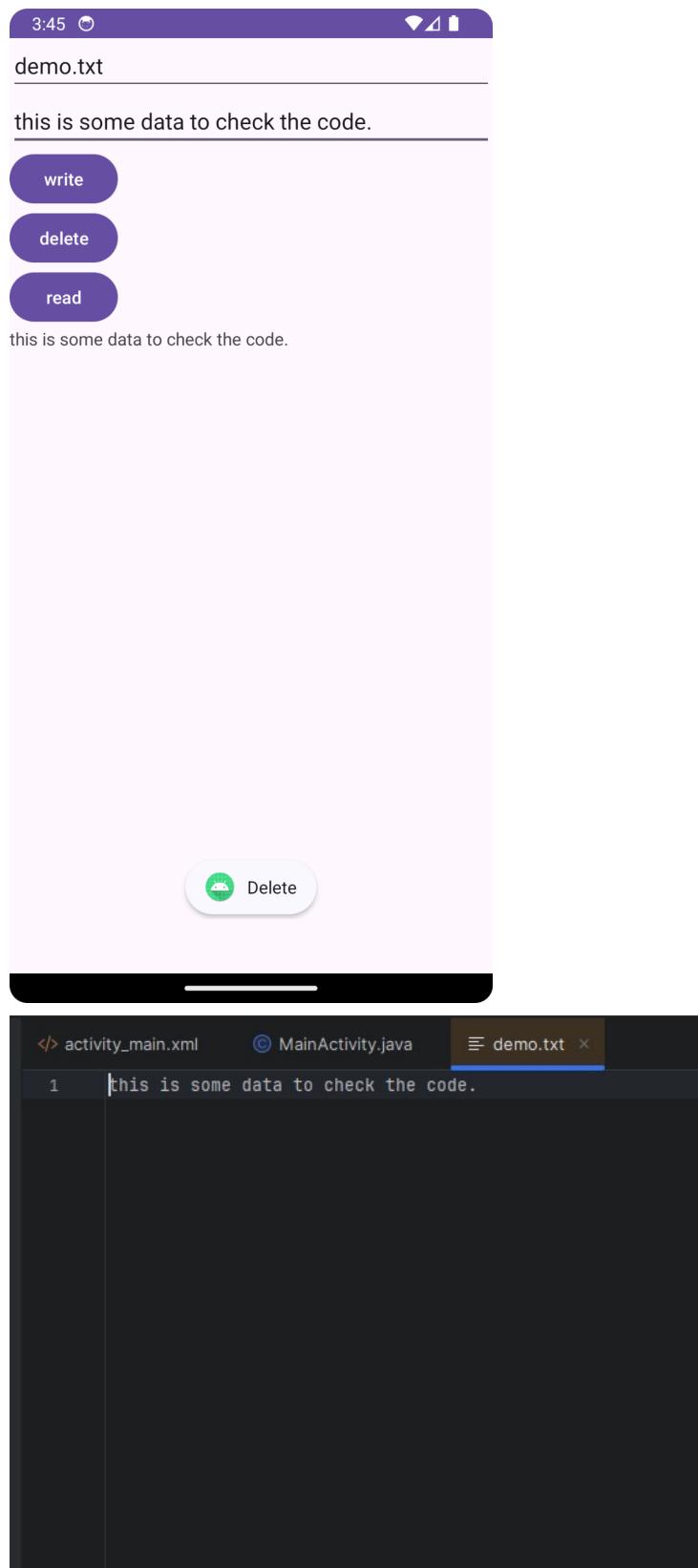
}) ;

delete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String f = fname.getText().toString();
        if(deleteFile(f)){
            Toast.makeText(MainActivity.this, "Delete",
Toast.LENGTH_SHORT).show();
        }
    }
}
e.printStackTrace();
```

Output:







6. Develop an app that allows users to capture and save media files (e.g., images, videos) to external storage. Implement functionality to capture a photo or video using the device's camera, save it to a specified directory on external storage, and provide options to share the media files using intents. Ensure that the app handles external storage permissions and provides feedback on successful or failed operations.

Solution:

Java file:

```
package com.example.myapplication;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.FileProvider;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import java.io.File;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_IMAGE_CAPTURE = 1;
    private static final int REQUEST_VIDEO_CAPTURE = 2;
    private Uri photoURI;
    private Uri videoURI;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button captureImageButton = findViewById(R.id.button_capture_image);
        Button captureVideoButton = findViewById(R.id.button_capture_video);
    }
}
```

```
captureImageButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        dispatchTakePictureIntent();
    }
}) ;

captureVideoButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        dispatchTakeVideoIntent();
    }
}) ;

// Request necessary permissions if not granted
ActivityCompat.requestPermissions(this,
        new String[]{Manifest.permission.CAMERA,
Manifest.permission.WRITE_EXTERNAL_STORAGE,
Manifest.permission.READ_EXTERNAL_STORAGE},
        1);
}

private void dispatchTakePictureIntent() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        File photoFile = null;
        try {
            photoFile = createImageFile();
        } catch (IOException ex) {
            Toast.makeText(this, "Error creating file",
Toast.LENGTH_SHORT).show();
        }
        if (photoFile != null) {
            photoURI = FileProvider.getUriForFile(this,
                getApplicationContext().getPackageName()
                ".fileprovider",
                photoFile);
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
            startActivityForResult(takePictureIntent,
REQUEST_IMAGE_CAPTURE);
        }
    }
}

private void dispatchTakeVideoIntent() {
```

```

Intent takeVideoIntent = new Intent(MediaStore.ACTION_VIDEO_CAPTURE) ;
if (takeVideoIntent.resolveActivity(getApplicationContext()) != null) {
    File videoFile = null;
    try {
        videoFile = createVideoFile();
    } catch (IOException ex) {
        Toast.makeText(this, "Error creating file",
Toast.LENGTH_SHORT).show();
    }
    if (videoFile != null) {
        videoURI = FileProvider.getUriForFile(this,
                getApplicationContext().getPackageName() +
".fileprovider",
                videoFile);
        takeVideoIntent.putExtra(MediaStore.EXTRA_OUTPUT, videoURI);
        startActivityForResult(takeVideoIntent, REQUEST_VIDEO_CAPTURE);
    }
}
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (resultCode == RESULT_OK) {
        if (requestCode == REQUEST_IMAGE_CAPTURE) {
            Toast.makeText(this, "Image saved to:\n" + photoURI.toString(),
Toast.LENGTH_LONG).show();
        } else if (requestCode == REQUEST_VIDEO_CAPTURE) {
            Toast.makeText(this, "Video saved to:\n" + videoURI.toString(),
Toast.LENGTH_LONG).show();
        }
    } else {
}
}

```

```

        Toast.makeText(this, "Operation failed",
Toast.LENGTH_SHORT).show();
    }
}
}

```

Xml file:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <Button
        android:id="@+id/button_capture_image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Capture Image" />

    <Button
        android:id="@+id/button_capture_video"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Capture Video" />
</LinearLayout>

```

File_paths.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<paths xmlns:android="http://schemas.android.com/apk/res/android"> <external-
    path name="external_files" path=". "/>
</paths>

```

Output:

7. Design an application that manages user profiles using SQLite. Create a database schema with tables for user information such as name, email, and profile picture. Implement functionality to add new profiles, update existing profiles, and delete profiles. Provide a user interface to display a list of profiles and allow users to interact with their data.

Solution:

JAVA FILE:

```

package com.example.assign;

import android.content.Context;

```

```
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.widget.*;
import android.view.*;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity {
```

```
    Button insert, delete, read, update;

    EditText title, content;
    TextView fdata;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        insert = findViewById(R.id.insert);
        delete = findViewById(R.id.delete);
        read = findViewById(R.id.read);
        update = findViewById(R.id.update);
        title = findViewById(R.id.title);
        content = findViewById(R.id.content);
        fdata = findViewById(R.id.fdata);

        dbHelper db = new dbHelper(MainActivity.this);
```

```

insert.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String t = title.getText().toString();
        String c = content.getText().toString();
        boolean f = db.insertData(t, c);
        if(f) {
            Toast.makeText(MainActivity.this, "Inserted",
Toast.LENGTH_SHORT).show();
        }
        else{
            Toast.makeText(MainActivity.this, "Not Inserted",
Toast.LENGTH_SHORT).show();
        }
    }
});

read.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Cursor res = db.getAllData(); if (res
!= null && res.getCount() > 0) {
            StringBuilder stringBuilder = new StringBuilder();
            while (res.moveToNext()) {
                String title = res.getString(0);
                String content = res.getString(1);
                stringBuilder.append("Title:
").append(title).append("\n")
.append("Content:
").append(content).append("\n\n");
            }
            fdata.setText(stringBuilder.toString());
        }
    }
});
delete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String titleToDelete = title.getText().toString().trim();
        if (!titleToDelete.isEmpty()) {
            int rowsDeleted = db.deleteData(titleToDelete);
            if (rowsDeleted > 0) {
                Toast.makeText(MainActivity.this, "Deleted",
Toast.LENGTH_SHORT).show();
            } else {

```

```

        Toast.makeText(MainActivity.this, "No items Deleted",
Toast.LENGTH_SHORT).show();
    }
}
} );
}

update.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String titleToUpdate = title.getText().toString().trim();
        String newContent = content.getText().toString().trim();

        if (!titleToUpdate.isEmpty() && !newContent.isEmpty()) {
            boolean isUpdated = db.updateData(titleToUpdate,
newContent);
            if (isUpdated) {
                Toast.makeText(MainActivity.this, "Updated",
Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(MainActivity.this, "No updates",
Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(MainActivity.this, "Please enter both title
and new content", Toast.LENGTH_SHORT).show();
        }
    }
}
}

```

XML FILE:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:orientation="vertical">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/title" />

```

```
    android:hint="enter name:"/>>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/content"
    android:hint="enter email:"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="insert"
    android:id="@+id/insert"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="delete"
    android:id="@+id/delete"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="read"
    android:id="@+id/read"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="update"
    android:id="@+id/update"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/fdata"/>
</LinearLayout>
```

```
package com.example.assign;
```

DATABASE HELPER FILE:

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class dbHelper extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = "Notes.db";
    public static final String TABLE_NAME = "NOTES";
    public static final String COL_1 = "TITLE";
    public static final String COL_2 = "CONTENT";

    public dbHelper(Context context) {
        super(context, DATABASE_NAME, null, 1);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table " + TABLE_NAME + " (" + COL_1 + " TEXT, " + COL_2 + " TEXT)");
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
    public boolean insertData(String t, String c) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_1, t);
        contentValues.put(COL_2, c);

        try {
            long result = db.insert(TABLE_NAME, null, contentValues);
            return result != -1;
        } catch (Exception e) {
            Log.e("DB_ERROR", "Error inserting data: " + e.getMessage());
            return false;
        }
    }
    public Cursor getAllData() {
        SQLiteDatabase db = this.getWritableDatabase();
```

```
        Cursor res = db.rawQuery(" select * from "+TABLE_NAME,null);
        db.rawQuery(" return res;

    }

    public boolean updateData(String t, String c) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_1,t);
        contentValues.put(COL_2,c);
        db.update(TABLE_NAME, contentValues, "TITLE = ?",new String[] { t });
        return true;
    }

    public Integer deleteData (String t) {
        SQLiteDatabase db = this.getWritableDatabase();
        return db.delete(TABLE_NAME, "TITLE = ?",new String[] {t});
    }
}
```

Output:

2:27



bob

bob@gmail.com

insert

delete

read

update



Caused by: o...



q¹ w² e³ r⁴ t⁵ y⁶ u⁷ i⁸ o⁹ p⁰

a s d f g h j k l

↶ z x c v b n m ↶

?123

,

Inserted

.

↶



8. Create an app that uses Shared Preferences to manage app-specific preferences such as theme selection (light/dark mode), font size, and language. Implement a settings screen to allow users to adjust these preferences and save their choices. Ensure that the app reflects the selected preferences throughout the app and persists them across app restarts.

Solution:

JAVA FILE:

```
package com.example.assign;

import android.content.SharedPreferences;
import android.os.Bundle;

import androidx.appcompat.app.*;
```

```
import androidx.core.view.*;

import android.view.View;
import android.widget.*;

public class MainActivity extends AppCompatActivity {

    ToggleButton dark;
    Sharedpreferences sp;
    Button submit;
    EditText fs, lang;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        sp = getSharedpreferences("prefs", MODE_PRIVATE);
        Sharedpreferences.Editor e = sp.edit();
        dark = findViewById(R.id.dark);
        fs = findViewById(R.id.fs);
        lang =
        findViewById(R.id.lang);
        submit = findViewById(R.id.submit);

        if(sp.getInt("dark", 0)==1) {
            dark.setChecked(true);
        }
        fs.setText(sp.getString("fontsize", ""));
        lang.setText(sp.getString("lang", ""));

        dark.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                if(isChecked) {
                    e.putInt("dark", 1);
                    e.apply();
                }
                else{
                    e.putInt("dark", 0);
                    e.apply();
                }
            }
        });
    }
}
```

```

    });

    submit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String fsize = fs.getText().toString();
            String l = lang.getText().toString();
            e.putInt("dark", 1);
            e.putString("fontsize", fsize);
            e.putString("lang", l);
            e.apply();
        }
    });
}
}

```

XML FILE:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Dark Mode:"/>

    <ToggleButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textOn="ON"
        android:textOff="OFF"
        android:id="@+id/dark"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="enter font size"
        android:id="@+id/fs"/>

    <EditText

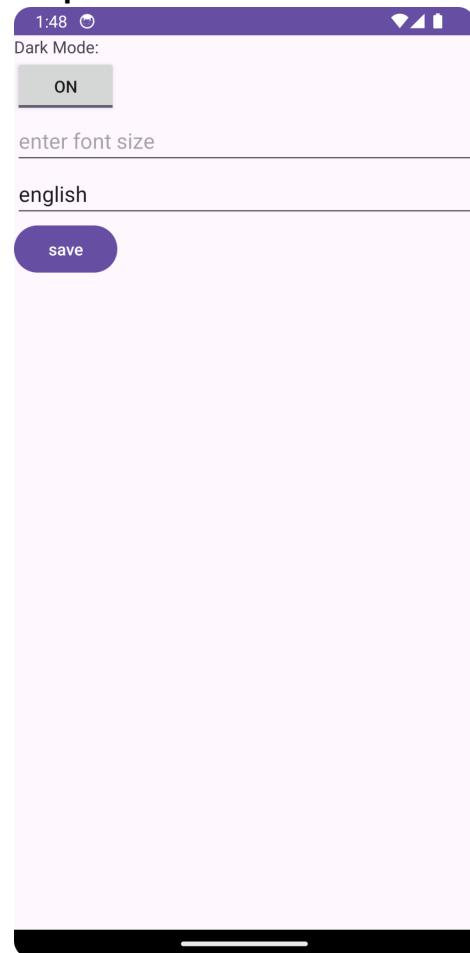
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="enter language"
        android:id="@+id/lang"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/submit"
        android:text="save"/>

</LinearLayout>
```

Output:



9. Develop an application that demonstrates data migration from Shared Preferences to SQLite. Start with an app that stores user preferences in Shared Preferences, and then migrate these preferences to a SQLite database. Implement functionality to read data from

Shared Preferences, insert it into the SQLite database, and ensure that the app continues to work with the new database.

Solution:

Java file:

```
package com.example.myapplication;

import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    private EditText editTextName, editTextAge;
    private Button buttonSave, buttonMigrate;
    private SharedPreferencesManager sharedpreferencesManager;
    private DatabaseHelper databaseHelper;

    private TextView tv;

    private static final String TAG = "MainActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editTextName = findViewById(R.id.editTextName);
        editTextAge = findViewById(R.id.editTextAge);
        buttonSave = findViewById(R.id.buttonSave);
        buttonMigrate = findViewById(R.id.buttonMigrate);
        tv = findViewById(R.id.tv);

        sharedpreferencesManager = new SharedPreferencesManager(this);
        databaseHelper = new DatabaseHelper(this);

        // Save Button Click Listener
        buttonSave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```

        String name = editTextName.getText().toString();
        String age = editTextAge.getText().toString();
        sharedpreferencesManager.saveUserData(name, age);
        Toast.makeText(MainActivity.this, "Data saved!",
Toast.LENGTH_SHORT).show();
    }
});

// Migrate Button Click Listener
buttonMigrate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        migrateData();
    }
});
}

private void migrateData() {
    SQLiteDatabase db = databaseHelper.getWritableDatabase();
    sharedpreferencesManager.migrateToSQLite(db);

    // Fetch and display data after migration
    fetchDataAndDisplay();
}

private void fetchDataAndDisplay() {
    String userData = databaseHelper.getUserData(); // Fetch data from
database if
    (!userData.isEmpty()) {
        tv.setText(userData);
    } else {
        Toast.makeText(MainActivity.this, "No data found!",
Toast.LENGTH_SHORT).show();
    }
}
}

```

DatabaseHelper:

```
package com.example.myapplication;

import android.annotation.SuppressLint;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "user_prefs.db";
    private static final int DATABASE_VERSION = 1;

    public static final String TABLE_USER_PREFS = "user_prefs";
    public static final String COLUMN_NAME = "name";
    public static final String COLUMN_AGE = "age";

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_TABLE = "CREATE TABLE " + TABLE_USER_PREFS + "("
                + COLUMN_NAME + " TEXT,"
                + COLUMN_AGE + " TEXT" + ")";
        db.execSQL(CREATE_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_USER_PREFS);
        onCreate(db);
    }

    public String getUserData() {
        SQLiteDatabase db = this.getReadableDatabase();
        String query = "SELECT * FROM " + TABLE_USER_PREFS;
        Cursor cursor = db.rawQuery(query, null);

        StringBuilder userData = new StringBuilder();

        if (cursor.moveToFirst()) {
            do {
                @SuppressLint("Range") String name =
                cursor.getString(cursor.getColumnIndex(COLUMN_NAME));
                @SuppressLint("Range") String age =
                cursor.getString(cursor.getColumnIndex(COLUMN_AGE));
                userData.append("Name: ").append(name).append(", Age: ")
                .append(age).append("\n");
            } while (cursor.moveToNext());
        }

        cursor.close();
        return userData.toString();
    }
}
```

```
}
```

SharedPreferencesManager:

```
package com.example.myapplication;

import android.content.Context; import
android.content.SharedPreferences;
import android.database.sqlite.SQLiteDatabase;

public class SharedPreferencesManager {
    private static final String PREFS_NAME = "user_prefs";
    private SharedPreferences sharedPreferences;

    public SharedPreferencesManager(Context context) {
        sharedPreferences = context.getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE);
    }

    public void saveUserData(String name, String age) {
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putString("name", name);
        editor.putString("age", age);
        editor.apply();
    }

    public void migrateToSQLite(SQLiteDatabase db) {
        String name = sharedPreferences.getString("name", null);
        String age = sharedPreferences.getString("age", null);

        if (name != null && age != null) {
            String insertQuery = "INSERT INTO " +
DatabaseHelper.TABLE_USER_PREFS +
                "(" + DatabaseHelper.COLUMN_NAME + ", " +
DatabaseHelper.COLUMN_AGE + ")"      VALUES" +
"(" + name + ", " + age + ")";
            db.execSQL(insertQuery);
            clearSharedPreferences();
        }
    }

    private void clearSharedPreferences() {
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.clear();
        editor.apply();
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/editTextName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Name" />

    <EditText
        android:id="@+id/editTextAge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Age" />

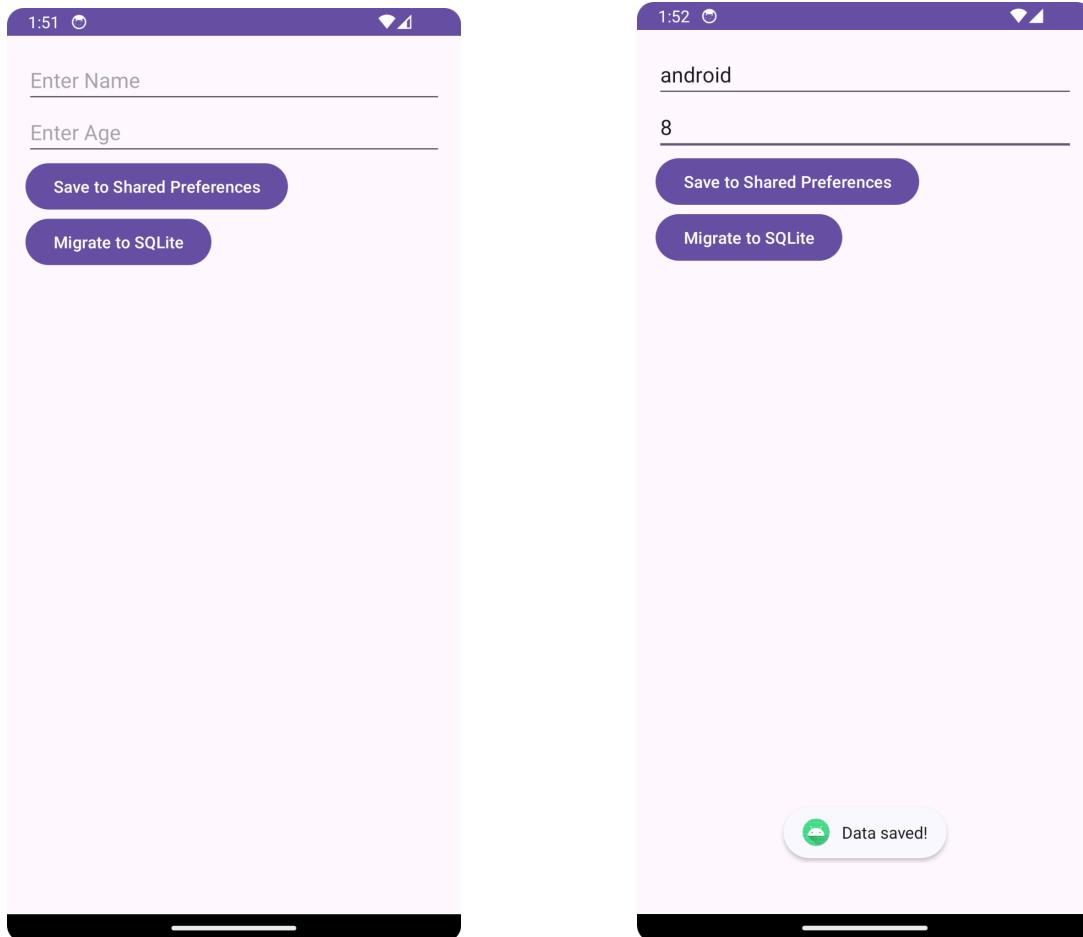
    <Button
        android:id="@+id/buttonSave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Save to Shared Preferences" />

    <Button
        android:id="@+id/buttonMigrate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Migrate to SQLite" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/tv"/>
</LinearLayout>
```

Xml file:

Output:



10. Design an app that includes a feature to backup and restore data. Use SQLite for storing app data and Shared Preferences for user settings. Implement functionality to create a backup file for the SQLite database and Shared Preferences data, store it in external storage, and provide an option to restore the data from the backup file. Ensure that the backup and restore operations handle errors and provide appropriate user feedback.

Solution:

```
package com.example.myapplication;

import android.content.SharedPreferences;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
```

```

public class MainActivity extends AppCompatActivity {
    private static final String PREFS_NAME = "UserSettings"; private
    DatabaseHelper dbHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        dbHelper = new DatabaseHelper(this);
        Button backupButton = findViewById(R.id.backupButton); Button
        restoreButton = findViewById(R.id.restoreButton);

        backupButton.setOnClickListener(new View.OnClickListener() { @Override

            public void onClick(View v) {
                BackupRestoreUtils.backupData(MainActivity.this);
                Toast.makeText(MainActivity.this, "Backup Successful",
Toast.LENGTH_SHORT).show();
            } });

        restoreButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                boolean success =
BackupRestoreUtils.restoreData(MainActivity.this);
                if (success) {
                    Toast.makeText(MainActivity.this, "Restore Successful",
Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(MainActivity.this, "Restore Failed",
Toast.LENGTH_SHORT).show();
                }
            } });
    }
}

```

BackupRestoreUtils.java:

```

package com.example.myapplication;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

```

```

public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "appdata.db";
    private static final int DATABASE_VERSION = 1;

    public DatabaseHelper(Context context) { super(context,
        DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // Create tables
        db.execSQL("CREATE TABLE user_data (id INTEGER PRIMARY
            KEY
AUTONINCREMENT,           name");
        TEXT)
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS");
        user_data
        onCreate(db);
    }
}

```

DatabaseHelper:

```

package com.example.myapplication;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "appdata.db";
    private static final int DATABASE_VERSION = 1;

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // Create tables
        db.execSQL("CREATE TABLE user_data (id INTEGER PRIMARY
            KEY
AUTONINCREMENT,           name");
        TEXT)
    }
}

```

```
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS");
    user_data
    onCreate(db);
}
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/backupButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Backup Data"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"/>

    <Button
        android:id="@+id/restoreButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Restore Data"
        android:layout_below="@+id/backupButton"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"/>
</RelativeLayout>
```

Xml file:

Output:

